Recursive Neural Networks (RNNs) for Semantic Relation Classification

SemEval 2010, Task 8

Multi-way classification of semantic relations between pairs of nominals
- Task setting: classifying relations between pairs of <e1, e2>
- Nine kinds of directed relations + “other” → 9 × 2 + 1 = 19 classes
- Examples:
  - Cause-Effect: The burst has been caused by the water hammer pressure
  - Instrument-Agency: This user plays games on a potable console

RNNs in Socher et al. (2012)
- Each word & node on parse trees: distributed vector
- Applying neural networks recursively on parse trees
- Performing softmax regression on any nodes of interest

Syntax-Based and Task-Specific Customization of RNNs

Utilizing Syntactic Information
- Phrase category-dependent composition functions
  - Composition functions may syntactically depend on components of phrases
  - Weighting head words and/or phrases
    - Head words and/or phrases are often important

Utilizing Task-Specific Information
- Weighting paths between pairs of target entities
  - For relation classification tasks, syntactic paths between target entities are important (Zhang+, 2006)
  - Vectors on the paths × β
  - Vectors on the others × (1 − β)
  - α ∈ [0, 1]

Averaging
- Averaging model parameters during L-BFGS optimization
  - The prediction performance often fluctuates significantly between training iterations
  - \( \hat{\theta} = \frac{1}{T+1} \sum_{t=0}^{T} \theta_t \) (\( \theta_t \): parameter vector at iteration t of L-BFGS optimization)

Classification Accuracy
- Comparison with other methods

<table>
<thead>
<tr>
<th>Method</th>
<th>F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>82</td>
</tr>
<tr>
<td>RNN</td>
<td>79</td>
</tr>
<tr>
<td>MV-RNN</td>
<td>80</td>
</tr>
<tr>
<td>MV-RNN w/POS,</td>
<td>81</td>
</tr>
<tr>
<td>NER</td>
<td>82</td>
</tr>
<tr>
<td>SVM w/ rich</td>
<td>81</td>
</tr>
</tbody>
</table>

More computationally expensive RNN model with word-dependent weight matrices
Using semantic external features

Contributions of proposed methods
- Generalized and stabilized well

<table>
<thead>
<tr>
<th>Method</th>
<th>F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours w/o PC</td>
<td>75</td>
</tr>
<tr>
<td>w/o Head</td>
<td>76</td>
</tr>
<tr>
<td>w/o Weight</td>
<td>77</td>
</tr>
<tr>
<td>w/o Path</td>
<td>76</td>
</tr>
<tr>
<td>w/o Average</td>
<td>78</td>
</tr>
<tr>
<td>w/o PC, HW, PW,</td>
<td>79</td>
</tr>
<tr>
<td>AVE</td>
<td>80</td>
</tr>
</tbody>
</table>

Effects of averaging
- L-BFGS vs mini-batch online learning (e.g. AdaGrad (Duchi+, 2011))
- Averaging vs dropout (Hinton+, 2012)
- Applying our methods to other datasets
- Recursive deep models with deeper syntactic information

Conclusions & Future Work
- Simple customization of RNNs
  - Classification accuracy improvement
- Averaging model parameters
  - Stabilized and well-generalized classification accuracy