Multiplicative Tree-Structured LSTMs

Weiwei Cheng
Amazon

joint work with Nam Khanh Tran
Distributed Word Representations

- Words are represented as real-valued vectors
- *E.g.*, Skip-Gram, GloVe
Distributed Sentence Representations

- Sentences are represented as real-valued vectors
- Useful in, e.g., sentence classification, natural language understanding
Sequence Models (RNN)

- Process sentence from left-to-right
- Input at each step is a word vector and previous hidden state
- Rightmost output is the representation of the sentence
- Common parameterization: RNN, LSTM
Recursive Tree Model

climbs

\[ \text{cat} \quad \text{the} \quad \text{tree} \quad \text{tall} \]

dependency tree
Recursive Tree Model

climbs

cat

the

tall

dependency tree

representation
TreeLSTMs [Tai et al. 2015]

\[ \tilde{h}_j = \sum_{k \in C(j)} h_k \]
\[ i_j = \sigma(W^{(i)}x_j + U^{(i)}\tilde{h}_j + b^{(i)}) \]
\[ o_j = \sigma(W^{(o)}x_j + U^{(o)}\tilde{h}_j + b^{(o)}) \]
\[ f_{jk} = \sigma(W^{(f)}x_j + U^{(f)}h_k + b^{(f)}) \]
\[ u_j = \tanh(W^{(u)}x_j + U^{(u)}\tilde{h}_j + b^{(u)}) \]
\[ c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k \]
\[ h_j = o_j \odot \tanh(c_j) \]
TreeLSTMs

- Generalization of the sequential LSTM composition function
- A separate forget gate for each child
- Selectively preserve information from each child
This Work

- We propose multiplicative TreeLSTM that utilizes not only the lexical information of words, but also the relation information between the words.

- We investigate the use of lexical semantic information induced by Abstract Meaning Representation in tree structured models.
A young girl is playing on the edge of a fountain and an older woman is not watching her.
Abstract Meaning Representation (AMR)

- AMR is a semantic formalism where the meaning of a sentence is encoded as a single rooted, directed, acyclic graph.
  
  - **AMR concepts**: predicate senses, named entity annotations, and lemmas.
  
  - **AMR relations**: semantic roles (Propbank), semantic relations defined specifically for AMR.
Tree-Structured LSTMs with AMR

Recursively apply $\phi$ on the AMR structure
Tree-Structured LSTMs with AMR

Recursively apply $\phi$ on the AMR structure
Tree-Structured LSTMs with AMR

- **Observation**: Distinct edges / relations between nodes → Possibility for flexible parametrization

- **Proposal**: Multiplicative TreeLSTMs for modeling relations
Our Contribution – Multiplicative TreeLSTM

- **Main idea:** Introduce fine-grained parameters based on the edge types, *i.e.*, separate transition matrix for each edge type.
Our Contribution – Multiplicative TreeLSTM

- **Main idea**: Introduce fine-grained parameters based on the edge types

\[
\begin{align*}
\tilde{h}_j &= \sum_{k \in C(j)} W_{hh}^{r(j,k)} h_k \\
i_j &= \sigma(W^{(i)}x_j + U^{(i)}\tilde{h}_j + b^{(i)}) \\
o_j &= \sigma(W^{(o)}x_j + U^{(o)}\tilde{h}_j + b^{(o)}) \\
f_{jk} &= \sigma(W^{(f)}x_j + U^{(f)}h_k + b^{(f)}) \\
u_j &= \tanh(W^{(u)}x_j + U^{(u)}\tilde{h}_j + b^{(u)}) \\
c_j &= i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k \\
h_j &= o_j \odot \tanh(c_j)
\end{align*}
\]
Our Contribution – Multiplicative TreeLSTM

- **Main idea:** Introduce fine-grained parameters based on the edge types.

- Use semantic relations to combine hidden states from children nodes, via $W_{hh}$.
  - But, 3-way tensors require large number of parameters → overfitting.

\[
\tilde{h}_j = \sum_{k \in C(j)} W_{hh}^{r(j,k)} h_k
\]
Our Contribution – Multiplicative TreeLSTM

- **Solution:** Factorize $W_{hh}^{r(j,k)}$ by using the product of two dense matrices shared across edge types, with an diagonal matrix that is edge-type dependent

$$W_{hh}^{r(j,k)} = W_{hm} \, \text{diag}(W_{mr} e_{jk}) \, W_{mh}$$

- The mapping $\tilde{h}_j = \sum_{k \in \mathcal{C}(j)} W_{hh}^{r(j,k)} h_k$ is then given by

$$m_{jk} = (W_{mr} e_{jk}) \odot (W_{mh} h_k)$$
$$\tilde{h}_j = \sum_{k \in \mathcal{C}(j)} W_{hm} m_{jk}$$
Our Contribution – Multiplicative TreeLSTM

- **mTreeLSTM** can be applied to any tree, where connection types between nodes are given, *e.g.*, dependency trees, AMR trees
  - Fewer number of parameters
  - Leverage potential correlation among fine-grained edge types
- In the experiments, we apply mTreeLSTM on a wide range of NLP tasks, including *sentiment classification, sentence relatedness*, and *natural language inference*, and investigate the usefulness of AMR.
Experiment 1: Sentiment Classification

- **Task**: Predict the sentiment of review sentences
  - Binary subtask: *positive* vs. *negative*
  - 5-class subtask: *strongly positive*, *positive*, *neutral*, *negative*, *strongly negative*

- **Data**: Stanford Sentiment Treebank [Socher et al. 2013]

- **Model**:
  - TreeLSTMs on given parse trees
  - Softmax classifier at root node
Experiment 1: Results

<table>
<thead>
<tr>
<th>model</th>
<th>fine-grained</th>
<th>binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>45.6</td>
<td>85.6</td>
</tr>
<tr>
<td>TreeLSTM (C)</td>
<td>46.3</td>
<td>85.8</td>
</tr>
<tr>
<td>TreeLSTM (D)</td>
<td>46.0</td>
<td>85.0</td>
</tr>
<tr>
<td>TreeLSTM (A)</td>
<td>44.4</td>
<td>82.9</td>
</tr>
<tr>
<td>mTreeLSTM (A)</td>
<td>45.2</td>
<td>83.2</td>
</tr>
<tr>
<td>mTreeLSTM (D)</td>
<td><strong>46.7</strong></td>
<td><strong>85.7</strong></td>
</tr>
</tbody>
</table>

Whenever a tree structure is applicable to both mTreeLSTM and TreeLSTM, the performance of mTreeLSTM with that tree structure is better.
Experiment 2: Sentence Relatedness

- **Task**: Predict how related two sentences are

- **Data**:
  - SICK from SemEval 2014 Task 1 [Marelli et al. 2014]
  - Manually annotated relatedness scores from 1 to 5

- **Model**:
  - TreeLSTMs on given parse trees
  - Relatedness scores predicted by an additional feedforward layer on top
## Experiment 2: Results

<table>
<thead>
<tr>
<th>model</th>
<th>Pearson</th>
<th>Spearman</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>0.8409</td>
<td>0.7782</td>
<td>0.3035</td>
</tr>
<tr>
<td>TreeLSTM (C)</td>
<td>0.8497</td>
<td>0.7904</td>
<td>0.2861</td>
</tr>
<tr>
<td>TreeLSTM (D)</td>
<td>0.8631</td>
<td>0.8034</td>
<td>0.2600</td>
</tr>
<tr>
<td>TreeLSTM (A)</td>
<td>0.8415</td>
<td>0.7742</td>
<td>0.2986</td>
</tr>
<tr>
<td>mTreeLSTM (A)</td>
<td>0.8527</td>
<td>0.7884</td>
<td>0.2788</td>
</tr>
<tr>
<td>mTreeLSTM (D)</td>
<td><strong>0.8717</strong></td>
<td><strong>0.8141</strong></td>
<td><strong>0.2443</strong></td>
</tr>
</tbody>
</table>

Dependency trees and mTreeLSTM work the best.
Experiment 3: Natural Language Inference

- **Task:** Predict relation of two sentences – *entailment, contradiction, neutral*

- **Data:**
  - SICK from SemEval 2014 Task 1
  - Stanford NLI dataset [Bowman et al. 2015]

- **Model:**
  - TreeLSTMs on given parse trees
  - Classes predicted by an additional feedforward layer on top
Experiment 3: SICK

<table>
<thead>
<tr>
<th>model</th>
<th>all</th>
<th>long sentence</th>
<th>negation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>77.3</td>
<td>74.6</td>
<td>77.5</td>
</tr>
<tr>
<td>TreeLSTM (C)</td>
<td>79.0</td>
<td>78.1</td>
<td>85.3</td>
</tr>
<tr>
<td>TreeLSTM (D)</td>
<td>82.9</td>
<td>81.0</td>
<td>84.3</td>
</tr>
<tr>
<td>TreeLSTM (A)</td>
<td>82.6</td>
<td>84.0</td>
<td>88.2</td>
</tr>
<tr>
<td>mTreeLSTM (A)</td>
<td>83.3</td>
<td>85.3</td>
<td>88.5</td>
</tr>
<tr>
<td>mTreeLSTM (D)</td>
<td>84.0</td>
<td>81.6</td>
<td>87.8</td>
</tr>
</tbody>
</table>
# Experiment 3: Model Size

<table>
<thead>
<tr>
<th>model</th>
<th># parameters</th>
<th>accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TreeLSTM (D)</td>
<td>301K</td>
<td>82.9</td>
</tr>
<tr>
<td>addTreeLSTM (D)</td>
<td>361K</td>
<td>83.4</td>
</tr>
<tr>
<td>fullTreeLSTM (D)</td>
<td>1.1M</td>
<td>83.5</td>
</tr>
<tr>
<td>mTreeLSTM (D)</td>
<td>361K</td>
<td>84.0</td>
</tr>
</tbody>
</table>
## Experiment 3: SNLI

<table>
<thead>
<tr>
<th>model</th>
<th>accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM [Bowman et al., 2015]</td>
<td>77.6</td>
</tr>
<tr>
<td>Syntax TreeLSTM [Yogatama et al., 2017]</td>
<td>80.5</td>
</tr>
<tr>
<td>CYK TreeLSTM [Maillard et al., 2017]</td>
<td>81.6</td>
</tr>
<tr>
<td>Gumbel TreeLSTM [Choi et al., 2018]</td>
<td>81.8</td>
</tr>
<tr>
<td>Gumbel TreeLSTM + leaf LSTM [Choi et al., 2018]</td>
<td>82.6</td>
</tr>
<tr>
<td>TreeLSTM (D)</td>
<td>81.0</td>
</tr>
<tr>
<td>mTreeLSTM (D)</td>
<td>81.9</td>
</tr>
</tbody>
</table>
Conclusions

- We present **multiplicative TreeLSTM** for distributed sentence representation that utilizes not only the lexical information of words, but also the **relation information** between the words.

- Modeling relation information is helpful: **mTreeLSTMs outperform TreeLSTMs** on the same parse trees.

- With **AMR** as backbone, tree structured models can effectively handle long-range and complex dependencies.
Multiplicative Tree-Structured LSTMs

Weiwei Cheng
Amazon

joint work with Nam Khanh Tran