Enhanced Long-Short Term Memory
Enhanced Local Inference
Premise – Hypothesis Problem

• Premise:
  • Several airlines polled saw costs grow more than expected, even after adjusting for inflation.

• Hypothesis:
  • Some of the companies in the poll reported cost increases.

• Relationship Label:
  • Entailment
Premise – Hypothesis Problem

• Premise:
  • A man wearing a white shirt and a blue jeans reading a newspaper while standing.

• Hypothesis:
  • A man is sitting down reading a newspaper.

• Relationship Label:
  • Contradiction
Enhanced LSTM Premise-Hypothesis Model
Input Encoding

\[ \tilde{a}_i = \text{BiLSTM}(a, i), \forall i \in [1, \ldots, \ell_a], \quad (1) \]

\[ \tilde{b}_j = \text{BiLSTM}(b, j), \forall j \in [1, \ldots, \ell_b]. \quad (2) \]
Long-Short Term Memory (LSTM)

- Basic LSTM Review
- Bi-Directional LSTM (BiLSTM)
- Tree-LSTM cell and structure.
Basic LSTM Cell
Basic LSTM Cell
Basic LSTM Cell Calculation

\[ i_t = \sigma(W_i x_t + U_i h_{t-1}), \]
\[ f_t = \sigma(W_f x_t + U_f h_{t-1}), \]
\[ o_t = \sigma(W_o x_t + U_o h_{t-1}), \]
\[ c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1}), \]
\[ h_t = o_t \odot \tanh(c_t), \]
Input Encoding BiLSTM

\[ \bar{a}_i = \text{BiLSTM}(a, i), \forall i \in [1, \ldots, \ell_a], \quad (1) \]

\[ \bar{b}_j = \text{BiLSTM}(b, j), \forall j \in [1, \ldots, \ell_b]. \quad (2) \]
Tree LSTM Cell
Tree LSTM Cell Calculation

\[ h_t = \text{TrLSTM}(x_t, h_{t-1}^L, h_{t-1}^R), \]  
(3)

\[ h_t = o_t \odot \tanh(c_t), \]  
(4)

\[ o_t = \sigma(W_o x_t + U_o^L h_{t-1}^L + U_o^R h_{t-1}^R), \]  
(5)

\[ c_t = f_t^L \odot c_{t-1}^L + f_t^R \odot c_{t-1}^R + i_t \odot u_t, \]  
(6)

\[ f_t^L = \sigma(W_f x_t + U_f^{LL} h_{t-1}^L + U_f^{LR} h_{t-1}^R), \]  
(7)

\[ f_t^R = \sigma(W_f x_t + U_f^{RL} h_{t-1}^L + U_f^{RR} h_{t-1}^R), \]  
(8)

\[ i_t = \sigma(W_i x_t + U_i^L h_{t-1}^L + U_i^R h_{t-1}^R), \]  
(9)

\[ u_t = \tanh(W_c x_t + U_c^L h_{t-1}^L + U_c^R h_{t-1}^R), \]  
(10)
Input Encoding

(a) \[ S \rightarrow \text{N} \rightarrow \text{VP} \rightarrow \text{V} \rightarrow \text{NP} \rightarrow \text{D} \rightarrow \text{N} \]

John hit the ball

(b) LSTM

John hit the ball
Enhanced LSTM Premise-Hypothesis Model
Enhanced LSTM Premise-Hypothesis Model

- Pre Encoding
- Input Encoding -
  - BiLSTM or TreeLSTM
- Local Inference Modeling
  - Weighted Attention Matrix
- Enhanced Local Inference Modeling
  - Augmented difference and dot product.
- Inference Composition
  - BiLSTM or TreeLSTM
- Prediction
  - Max & Average or Max & Average & TreeRoot
  - SoftMax
Enhanced LSTM Model Pre Encoding

- Pre-trained 300-D Glove 840B vector word encoding.

- Syntax Parse Tree of 300/600D? encoded words.
  - Each node in the parse tree is an input to the model.

- Constituency parser produces the tree structures for both the premise and hypothesis.
Constituent Parse Tree

Figure 2: Head-Lexicalized Constituent Tree.
Premise Parse Tree

(a) Binarized constituency tree of premise
(b) Binarized constituency tree of hypothesis

A man is sitting down reading a newspaper.
Enhanced LSTM Model Input Encoding

- Each node in the parse tree is encoding as a word.
- The in-order traversal of the tree is the input vector.
- BiLSTM Model or TreeLSTM Model
- All xLSTM hidden states have 300 dimensions
  - (to match the 300-D encodings?)
- Unclear on number of cells in BiLSTM or TreeLSTM.
- All hidden states from LSTM cells collected for inference modeling at the next layer.
\[
\bar{a}_i = \text{BiLSTM}(a, i), \forall i \in [1, \ldots, \ell_a], \quad (1)
\]
\[
\bar{b}_j = \text{BiLSTM}(b, j), \forall j \in [1, \ldots, \ell_b]. \quad (2)
\]

OR

\[
h_t = \text{TrLSTM}(x_t, h_{t-1}^L, h_{t-1}^R), \quad (3)
\]
Enhanced LSTM Premise-Hypothesis Model
Local Inference Modeling

- Attention Matrix
- Weighted Attention Matrix
- Enhancing Inference Extraction
Attention Matrix

- All the hidden states of the BiLSTM/TreeLSTM cells for the premise are concatenated into vector \( A \).
- All the hidden states of the BiLSTM/TreeLSTM cells for the hypothesis are concatenated into vector \( B \).
- Calculate the attention matrix:

\[
e_{ij} = \overline{a_i}^T \overline{b_j}. \quad (11)
\]

\[
e_{ij} = \overline{a_i}^T \overline{b_j}, \forall i \in [1, \ldots, \ell_a], \forall j \in [1, \ldots, \ell_b].
\]
Weighted Attention Matrix

- The attention matrix represents how associated any two words <Ai,Bj> in the hypothesis and premise are.
- Compute A’, A weighted against B and vice versa, B’.
- For a given word Ai
  - Sum the attention from all <Ai, Bx>, as the weight.
  - Each attention <Ai,Bx> is divided by weight, as the scale
  - Sum each scale multiplied by the associated Bx.
Weighted Attention Matrix

\[ \tilde{a}_i = \sum_{j=1}^{\ell_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_b} \exp(e_{ik})} \bar{b}_j, \forall i \in [1, \ldots, \ell_a], \quad (12) \]

\[ \tilde{b}_j = \sum_{i=1}^{\ell_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_a} \exp(e_{kj})} \bar{a}_i, \forall j \in [1, \ldots, \ell_b], \quad (13) \]
Attention Matrix

- The matrix for BiLSTM and TreeLSTM is different as the TreeLSTM retains syntactic parse tree information in the internal nodes.
- Once the weighted attention matrix is computed then vector A’ and B’ can be solved.
- Vector A’ and B’ are normalized before enhancement.
Weighted Attention Matrix for BiLSTM

(f) Normalized attention weights of BiLSTM
Weighted Attention Vector - BiLSTM

(e) Input gate of BiLSTM in *inference composition* ($l^2$-norm)
Weighted Attention Matrix for TreeLSTM

(c) Normalized attention weights of tree-LSTM
(d) Input gate of tree-LSTM in *inference composition* ($l^2$-norm)
Enhanced Sequential Inference Extraction

- The vectors $A/B$ are all the hidden states from the input encoding layer.
- The vectors $A'/B'$ are the weighted inference calculated from the hidden states.
- Additional information can be extracted by taking the difference of $A-A'$ and the element wise product $A*A'$.
- Concatenate all 4 vectors into one new vector:
Enhancing Inference Extraction

\[ m_a = [\bar{a}; \tilde{a}; \bar{a} - \tilde{a}; \bar{a} \odot \tilde{a}], \quad (14) \]

\[ m_b = [\bar{b}; \tilde{b}; \bar{b} - \tilde{b}; \bar{b} \odot \tilde{b}]. \quad (15) \]
Enhanced LSTM Premise-Hypothesis Model
Inference Composition

- Enhanced Inference Extraction expands input size 4x.
- After EIE computation use a 1 layer feed forward NN to reduce the output
- BiLSTM and TreeLSTM are used again, respectively.
- Some changes to BiLSTM/TreeLSTM, as the original input was $X_t$ and now the input is now $F(M_a,t)$. (1-NN output)

\begin{align*}
v_{a,t} &= \text{TrLSTM}(F(m_{a,t}), h^L_{t-1}, h^R_{t-1}), \quad (16) \\
v_{b,t} &= \text{TrLSTM}(F(m_{b,t}), h^L_{t-1}, h^R_{t-1}). \quad (17)
\end{align*}
Enhanced LSTM Premise-Hypothesis Model
Poolings: Averaging, Max, and Root.

- **Average Pooling**: Take the average of the vector.
- **Max Pooling**: Take the max of the vector.
- **Root Pooling**: Take the root of a TreeLSTM tree.
- **Compose a new vector of the concatenated pools**

\[
\textbf{v}_{a,\text{ave}} = \sum_{i=1}^{\ell_a} \frac{\textbf{v}_{a,i}}{\ell_a}, \quad \textbf{v}_{a,\text{max}} = \max_{i=1}^{\ell_a} \textbf{v}_{a,i}, \quad \textbf{v}_{b,\text{ave}} = \sum_{j=1}^{\ell_b} \frac{\textbf{v}_{b,j}}{\ell_b}, \quad \textbf{v}_{b,\text{max}} = \max_{j=1}^{\ell_b} \textbf{v}_{b,j}, \\
\textbf{v} = [\textbf{v}_{a,\text{ave}}; \textbf{v}_{a,\text{max}}; \textbf{v}_{b,\text{ave}}; \textbf{v}_{b,\text{max}}].
\]
The concatenated vector is fed into a multi-layer perceptron classifier.

- Hidden layer: tanh
- Output layer: softmax
### ESIM & Hybrid Inference Model (HIM) - Results

<table>
<thead>
<tr>
<th>Model</th>
<th>#Para.</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Handcrafted features (Bowman et al., 2015)</td>
<td>-</td>
<td>99.7</td>
<td>78.2</td>
</tr>
<tr>
<td>(2) 300D LSTM encoders (Bowman et al., 2016)</td>
<td>3.0M</td>
<td>83.9</td>
<td>80.6</td>
</tr>
<tr>
<td>(3) 1024D pretrained GRU encoders (Vendrov et al., 2015)</td>
<td>15M</td>
<td>98.8</td>
<td>81.4</td>
</tr>
<tr>
<td>(4) 300D tree-based CNN encoders (Mou et al., 2016)</td>
<td>3.5M</td>
<td>83.3</td>
<td>82.1</td>
</tr>
<tr>
<td>(5) 300D SPINN-PI encoders (Bowman et al., 2016)</td>
<td>3.7M</td>
<td>89.2</td>
<td>83.2</td>
</tr>
<tr>
<td>(6) 600D BiLSTM intra-attention encoders (Liu et al., 2016)</td>
<td>2.8M</td>
<td>84.5</td>
<td>84.2</td>
</tr>
<tr>
<td>(7) 300D NSE encoders (Munkhdalai and Yu, 2016a)</td>
<td>3.0M</td>
<td>86.2</td>
<td>84.6</td>
</tr>
<tr>
<td>(8) 100D LSTM with attention (Rocktäschel et al., 2015)</td>
<td>250K</td>
<td>85.3</td>
<td>83.5</td>
</tr>
<tr>
<td>(9) 300D mLSTM (Wang and Jiang, 2016)</td>
<td>1.9M</td>
<td>92.0</td>
<td>86.1</td>
</tr>
<tr>
<td>(10) 450D LSTMN with deep attention fusion (Cheng et al., 2016)</td>
<td>3.4M</td>
<td>88.5</td>
<td>86.3</td>
</tr>
<tr>
<td>(11) 200D decomposable attention model (Parikh et al., 2016)</td>
<td>380K</td>
<td>89.5</td>
<td>86.3</td>
</tr>
<tr>
<td>(12) Intra-sentence attention + (11) (Parikh et al., 2016)</td>
<td>580K</td>
<td>90.5</td>
<td>86.8</td>
</tr>
<tr>
<td>(13) 300D NTI-SLSTM-LSTM (Munkhdalai and Yu, 2016b)</td>
<td>3.2M</td>
<td>88.5</td>
<td>87.3</td>
</tr>
<tr>
<td>(14) 300D re-read LSTM (Sha et al., 2016)</td>
<td>2.0M</td>
<td>90.7</td>
<td>87.5</td>
</tr>
<tr>
<td>(15) 300D btree-LSTM encoders (Paria et al., 2016)</td>
<td>2.0M</td>
<td>88.6</td>
<td>87.6</td>
</tr>
<tr>
<td>(16) 600D ESIM</td>
<td>4.3M</td>
<td>92.6</td>
<td>88.0</td>
</tr>
<tr>
<td>(17) HIM (600D ESIM + 300D Syntactic tree-LSTM)</td>
<td>7.7M</td>
<td>93.5</td>
<td><strong>88.6</strong></td>
</tr>
</tbody>
</table>
## Ablation Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>(17) HIM (ESIM + syn.tree)</td>
<td>93.5</td>
<td>88.6</td>
</tr>
<tr>
<td>(18) ESIM + tree</td>
<td>91.9</td>
<td>88.2</td>
</tr>
<tr>
<td>(16) ESIM</td>
<td>92.6</td>
<td>88.0</td>
</tr>
<tr>
<td>(19) ESIM - ave./max</td>
<td>92.9</td>
<td>87.1</td>
</tr>
<tr>
<td>(20) ESIM - diff./prod.</td>
<td>91.5</td>
<td>87.0</td>
</tr>
<tr>
<td>(21) ESIM - inference BiLSTM</td>
<td>91.3</td>
<td>87.3</td>
</tr>
<tr>
<td>(22) ESIM - encoding BiLSTM</td>
<td>88.7</td>
<td>86.3</td>
</tr>
<tr>
<td>(23) ESIM - P-based attention</td>
<td>91.6</td>
<td>87.2</td>
</tr>
<tr>
<td>(24) ESIM - H-based attention</td>
<td>91.4</td>
<td>86.5</td>
</tr>
<tr>
<td>(25) syn.tree</td>
<td>92.9</td>
<td>87.8</td>
</tr>
</tbody>
</table>

Table 2: Ablation performance of the models.
Questions?