Adversarial Connective-exploiting Networks for Implicit Discourse Relation Classification

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, Eric P. Xing

Shubham Jain
Discourse Relations

• Connect linguistic units (like sentences) semantically

• Types:
  • Explicit:
    *I like the food, but I am full.* (Relation: Comparison)
  
  Use Connectives

  • Implicit:
    *Never mind. You already know the answer.*

  Connectives can be inferred
Implicit discourse relation

Units: Never mind. You already know the answer.

Connective: Never mind. *Because* you already know the answer.

Sentence 1: Never mind.
Sentence 2: You already know the answer.

*Implicit connective*: Because
*Discourse relation*: Cause
Discourse relation Classification

- Connectives are very important cues
- Explicit discourse relation: > 85%
- Implicit discourse relation: < 50% (with end to end neural nets !!!)

\[ x_1: \text{Never mind.} \]
\[ x_2: \text{You Know the answer.} \]

\[ \text{NN} \rightarrow H^i \rightarrow \text{Classifier C} \rightarrow \text{Relation: Cause} \]
The Idea

- Human annotators adds the connectives to the dataset to find the relation

- Example from Penn Discourse Treebank (PDTB) benchmark
  
  Never mind. You already know the answer.

  - Add the implicit *connective*
    
    Never mind. *because* You already know the answer.

  - Determine the relation
Idea

• Use the annotated implicit connectives in the training data

\[ x_1: \text{Never mind.} \]
\[ x_2: \text{You Know the answer.} \]

+implicit connective $c$: **Because**

\[ x_1: \text{Never mind.} \]
\[ x_2: \textbf{Because} \text{ You Know the answer.} \]

Implicit feature

Relation: Cause

Imitates the connective-augmented feature to improve discriminability

Relation: Cause

Highly-discriminative connective-augmented feature for classification
Feature imitation

• Due to the connective cue, there is a huge gap in the features

• Failed with using things like L2 distance reduction

• It was necessary to use adaptive scheme to ensure discriminability: Adversarial networks
Adversarial Networks

• Proposed by Goodfellow et al., 2014

• Idea:
  Say we want to generate images from a vector.
  • **Generator**: generate similar to a “correct values” to fool the discriminator
  • **Discriminator**: discriminate between the thing generated by the generator and the actual “correct values”
The model

- $x_1$: Never mind.
- $x_2$: You Know the answer.
- $x_1$: Never mind.
- $x_2$: Because You Know the answer.

- \textbf{i-CNN} wants to mimic \textbf{a-CNN} and both wants to maximize the classification accuracy from C
- Discriminator wants to discriminate between $H_{i}$ and $H_{A}$
Network training

Repeat :

• Train $i$-CNN and $C$ to maximize classification accuracy and fool $D$
• Train $a$-CNN to maximize classification accuracy
• Train $D$ to distinguish between the two features

Note: $a$-CNN is trained with $C$ fixed as it is strong enough
Network details: CNNs

• **i-CNN**
  - Word - Embedding layers, Convolutions and max-pooling

• **a-CNN**
  - Word - Embedding layers, Convolutions
  - Average k-max pooling
    - Average of the top k values
    - Forces to “attend” the contextual features from the sentences
Network details: Discriminator

• **Discriminator, D:**
  - Multi fully connected layers (FCs)
  - Additional stacked gate to help in gradient propagation [Qin et al., 2016]

• **Classifier, C:**
  - Fully connected layer followed by softmax
Experiments

• PDTB benchmark dataset
  • Sentence pairs, relation labels, implicit connectives

• Multi-class classification task
  • 11 relation classes
  • Two slightly different settings as in previous work

• One-vs-all classification tasks
  • 4 Relation classes: Comparison, Contingency, Expansion, Temporal
Multi-class classification task

- Accuracy (%) on two settings

<table>
<thead>
<tr>
<th>Model</th>
<th>PDTB-Lin</th>
<th>PDTB-Ji</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Word-vector</td>
<td>34.07</td>
<td>36.86</td>
</tr>
<tr>
<td>2 CNN</td>
<td>43.12</td>
<td>44.51</td>
</tr>
<tr>
<td>3 Ensemble</td>
<td>42.17</td>
<td>44.27</td>
</tr>
<tr>
<td>4 Multi-task</td>
<td>43.73</td>
<td>44.75</td>
</tr>
<tr>
<td>5 $\ell_2$-reg</td>
<td>44.12</td>
<td>45.33</td>
</tr>
<tr>
<td>6 Lin et al. (2009)</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>7 Lin et al. (2009)+Brown clusters</td>
<td></td>
<td>40.66</td>
</tr>
<tr>
<td>8 Ji and Eisenstein (2015)</td>
<td></td>
<td>44.59</td>
</tr>
<tr>
<td>9 Qin et al. (2016a)</td>
<td></td>
<td>45.04</td>
</tr>
<tr>
<td>10 Ours</td>
<td>44.65</td>
<td>46.23</td>
</tr>
</tbody>
</table>
One-vs-all classification tasks

- Comparisons of F1 scores (%) for binary classifications

<table>
<thead>
<tr>
<th>Model</th>
<th>COMP.</th>
<th>CONT.</th>
<th>EXP.</th>
<th>TEMP.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitler et al. (2009)</td>
<td>21.96</td>
<td>47.13</td>
<td>-</td>
<td>16.76</td>
</tr>
<tr>
<td>Qin et al. (2016c)</td>
<td><strong>41.55</strong></td>
<td><strong>57.32</strong></td>
<td>71.50</td>
<td>35.43</td>
</tr>
<tr>
<td>Zhang et al. (2016)</td>
<td>35.88</td>
<td>50.56</td>
<td>71.48</td>
<td>29.54</td>
</tr>
<tr>
<td>Zhou et al. (2010)</td>
<td>31.79</td>
<td>47.16</td>
<td>70.11</td>
<td>20.30</td>
</tr>
<tr>
<td>Liu and Li (2016)</td>
<td>36.70</td>
<td>54.48</td>
<td>70.43</td>
<td><strong>38.84</strong></td>
</tr>
<tr>
<td>Chen et al. (2016a)</td>
<td>40.17</td>
<td>54.76</td>
<td>-</td>
<td>31.32</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>40.87</td>
<td>54.56</td>
<td><strong>72.38</strong></td>
<td>36.20</td>
</tr>
</tbody>
</table>
Feature visualization

• *i*-CNN (blue) and *a*-CNN (orange) feature vectors
  • (a): without adversarial mechanism
  • (b)-(c): features as training proceeds in the proposed framework
Conclusions

• Connectives are very important cues

• Use the additional data during training to propose a new feature learning

• Proposed adversarial networks for feature learning with adaptive distance
Discussions

• Generalization
  • Can be used in task in which we can use additional data during training time to learn better
Thanks