BiRRE: Learning Bidirectional Residual Relation Embeddings for Supervised Hypernymy Detection

Chengyu Wang\textsuperscript{1,2}, Xiaofeng He\textsuperscript{3}\textsuperscript{*}

\textsuperscript{1} School of Software Engineering, East China Normal University, Shanghai, China
\textsuperscript{2} Alibaba Group, Hangzhou, China
\textsuperscript{3} School of Computer Science and Engineering, East China Normal University, Shanghai, China
Outline

- Introduction
- The BiRRE Model
  - M1: Hyponym Projection
  - M2: Hypernym Projection
  - M3: Hypernymy Relation Classification
- Experiments
- Conclusion
Introduction (1)

- **Hypernymy** ("is-a") relations are important for NLP and Web applications
  - Semantic resource construction: semantic hierarchies, taxonomies, knowledge graphs, etc.
  - Web-based applications: query understanding, post-search navigation, personalized recommendation, etc.

- **Predicting hypernymy relations between term pairs**
  - Pattern-based approaches: have low recall
  - Distributional classifiers: suffer from the “lexical memorization” problem
Introduction (2)

• **Our Idea: Learning Bidirectional Residual Relation Embeddings**
  
  – High performance: distributional models
  – Alleviating the “lexical memorization” problem: avoiding classifying hypernymy vs. non-hypernymy relations using word vectors as features directly
  
  – Two ways of modeling the hypernymy relations:
    • Hyponym projection: mapping hypernyms to hyponyms in the embedding space
    • Hypernym projection: mapping hyponyms to hypernyms in the embedding space
  
  – Model design: given a term pair \((x, y)\), measuring whether
    
    • \(\bar{x}\) can be projected to \(\bar{y}\) by hypernym projection
    • \(\bar{y}\) can be projected to \(\bar{x}\) by hyponym projection

    Positive sample: (cat, mammal)
    Negative sample: (desk, fruit)
BiRRE: The Proposed Framework

M1: Hyponym Projection
- \( \text{hypo}^{(1)}(y_i) \)
- \( \text{hypo}^{(2)}(y_i) \)
- \( \text{hypo}^{(N)}(y_i) \)

M2: Hypernym Projection
- \( \text{hyper}(x_i) \)
- \( \text{res}^{\text{hyper}}(x_i, y_i) \)

M3: Hypernymy Relation Classification
- Training
- Regularization

Hypernymy & Non-hypernymy Relations \( D^{(+)} \cup D^{(-)} \)

Pre-processing

Term Pairs

Embedding Lookup

Classifier

BiRRE Vector
Hidden Layers
Hyponym Projection (M1)

- **Learning $N$ projection matrices from hypernyms to hyponyms**
  
  - Simple objective function
    \[
    \min_{\mathcal{M}} \sum_{(x_i, y_i) \in D(+)} \sum_{p=1}^{N} \theta_i^{(p)} \|M^{(p)} y_i - x_i\|^2
    \]
    \[
    \text{s. t. } M^{(p)^T} M^{(p)} = I_d, p \in \{1, \cdots, N\}
    \]
  
  - Considering negative regularization
    \[
    \min_{\mathcal{M}} \frac{1}{|D(+)|} \sum_{(x_i, y_i) \in D(+)} \sum_{p=1}^{N} \theta_i^{(p)} \|M^{(p)} y_i - x_i\|^2
    \]
    \[
    + \frac{\lambda}{|D(-)|} \sum_{(x_i, y_i) \in D(-)} \sum_{p=1}^{N} \phi_i^{(p)} (M^{(p)} y_i)^T \cdot x_i
    \]
    \[
    \text{s. t. } M^{(p)^T} M^{(p)} = I_d, p \in \{1, \cdots, N\}
    \]

- No standard off-the-shelf learning algorithm!
Hyponym Projection (M1)

- Efficient learning algorithm for hyponym projection
  - Slight changes of the objective function
    \[
    \min_{M} \frac{1}{|D(+)|} \sum_{(x_i, y_i) \in D(+)} \sum_{p=1}^{N} \theta_i^{(p)} \|M^{(p)}y_i - x_i\|^2 \\
    - \frac{\lambda}{|D(-)|} \sum_{(x_i, y_i) \in D(-)} \sum_{p=1}^{N} \phi_i^{(p)} \|M^{(p)}y_i - x_i\|^2
    \]
    s. t. \( M^{(p)T}M^{(p)} = I_d, p \in \{1, \ldots, N\} \)

- Learning projection matrices
  1: for \( p = 1 \) to \( N \) do
  2: \( B^{(p)} = \sum_{(x_i, y_i) \in D(+)} \theta_i^{(p)} x_i y_i^T \\
    \quad - \alpha \cdot \sum_{(x_i, y_i) \in D(-)} \phi_i^{(p)} x_i y_i^T; \)
  3: \( U^{(p)} \Sigma^{(p)} V^{(p)T} = \text{SVD}(B^{(p)}); \)
  4: \( R^{(p)} = \text{diag}(1, \ldots, 1, \det(U^{(p)})\det(V^{(p)})); \)
  5: \( M^{(p)} = U^{(p)} R^{(p)} V^{(p)T}; \)
  6: end for

- Learning latent variables
  \[
  \theta_i^{(p)*} = \theta_i^{(p)} - \eta \cdot \sum_{(x_i, y_i) \in D(+)} \|M^{(p)}y_i - x_i\|^2 \\
  \phi_i^{(p)*} = \phi_i^{(p)} + \eta \cdot \sum_{(x_i, y_i) \in D(-)} \|M^{(p)}y_i - x_i\|^2
  \]
  * Refer to the proof of correctness in the paper.
Hypernym Projection (M2) &
Hypernymy Relation Classification (M3)

• Learning one projection matrix from hypernyms to hyponyms
  – Objective function

\[
\min_Q \frac{1}{|D^-|} \sum_{(x_i, y_i) \in D^-} \|Qx_i - y_i\|^2 - \frac{\lambda}{|D^+|} \sum_{(x_i, y_i) \in D^+} \|Qx_i - y_i\|^2 \text{ s. t. } Q^TQ = I_d
\]
  – Learning algorithm: a simpler version of M1

• Training of the hypernymy relation classifier
  – Hyponym residual vector: \( res_{\text{hypo}}(x_i, y_i) = x_i - M^{(\tilde{p})}y_i \)
  – Hypernym residual vector: \( res_{\text{hyper}}(x_i) = Qx_i - y_i \)
  – Feature representations: \( r_i = res_{\text{hypo}}(x_i, y_i) \oplus res_{\text{hyper}}(x_i, y_i) \)
  – Classifier learning: simple back propagation training of feed-forward neural networks
Experiments (1)

- **Experimental Settings**
  - Word embeddings: fastText embeddings, \( d = 300 \)
  - Default parameters settings:
    - \( \eta = 0.001, N = \max\{1, |\lg D^+|\} \)
  - Optimization: Adam with dropout rate 0.1

- **Effectiveness of BiRRE over the largest dataset (Shwartz et al. 2016)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision Random Split</th>
<th>Recall Random Split</th>
<th>F1 Random Split</th>
<th>Precision Lexical Split</th>
<th>Recall Lexical Split</th>
<th>F1 Lexical Split</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roller and Erk (2016)</td>
<td>0.926</td>
<td>0.850</td>
<td>0.886</td>
<td>0.700</td>
<td>0.964</td>
<td>0.811</td>
</tr>
<tr>
<td>Shwartz et al. (2016)</td>
<td>0.913</td>
<td>0.890</td>
<td>0.901</td>
<td>0.809</td>
<td>0.617</td>
<td>0.700</td>
</tr>
<tr>
<td>Glavas and Ponzetto (2017)</td>
<td>0.933</td>
<td>0.826</td>
<td>0.876</td>
<td>0.705</td>
<td>0.785</td>
<td>0.743</td>
</tr>
<tr>
<td>Rei et al. (2018)</td>
<td>0.928</td>
<td>0.887</td>
<td>0.907</td>
<td>0.826</td>
<td>0.860</td>
<td>0.842</td>
</tr>
<tr>
<td><strong>BiRRE</strong></td>
<td><strong>0.945</strong></td>
<td><strong>0.932</strong></td>
<td><strong>0.938</strong></td>
<td><strong>0.880</strong></td>
<td>0.918</td>
<td><strong>0.898</strong></td>
</tr>
</tbody>
</table>
Experiments (2)

• **General Performance**
  – Results over two general benchmark datasets
    • BLESS
    • ENTAILMENT

<table>
<thead>
<tr>
<th>Method</th>
<th>BLESS</th>
<th>ENT.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mikolov et al. (2013)</td>
<td>0.84</td>
<td>0.83</td>
</tr>
<tr>
<td>Yu et al. (2015)</td>
<td>0.90</td>
<td>0.87</td>
</tr>
<tr>
<td>Luu et al. (2016)</td>
<td>0.93</td>
<td>0.91</td>
</tr>
<tr>
<td>Nguyen et al. (2017)</td>
<td>0.94</td>
<td>0.91</td>
</tr>
<tr>
<td>Wang et al. (2019a)</td>
<td>0.97</td>
<td>0.92</td>
</tr>
<tr>
<td><strong>BiRRE</strong></td>
<td><strong>0.98</strong></td>
<td><strong>0.93</strong></td>
</tr>
</tbody>
</table>

• **Ablation Study**
  – Choice of baselines
    • Addition, offset and concatenation of term vectors
    • Unidirectional residual vectors

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>BLESS</th>
<th>ENT.</th>
<th>Shwartz</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_i + y_i$</td>
<td>0.76</td>
<td>0.77</td>
<td>0.72</td>
</tr>
<tr>
<td>$x_i - y_i$</td>
<td>0.79</td>
<td>0.74</td>
<td>0.73</td>
</tr>
<tr>
<td>$x_i \oplus y_i$</td>
<td>0.81</td>
<td>0.80</td>
<td>0.77</td>
</tr>
<tr>
<td>$\text{res}^{\text{hypo}}(x_i, y_i)$</td>
<td>0.92</td>
<td>0.87</td>
<td>0.84</td>
</tr>
<tr>
<td>$\text{res}^{\text{hyper}}(x_i, y_i)$</td>
<td>0.89</td>
<td>0.84</td>
<td>0.82</td>
</tr>
<tr>
<td>$r_i$ (i.e., BiRRE)</td>
<td><strong>0.99</strong></td>
<td><strong>0.93</strong></td>
<td><strong>0.88</strong></td>
</tr>
</tbody>
</table>

* Refer to more experiments in the paper.
Conclusion

• **Model**
  – A distributional model for supervised hypernymy detection based on bidirectional residual relation embeddings

• **Results**
  – BiRRE outperforms previous strong baselines over various evaluation frameworks

• **Future Work**
  – Improving projection learning to model complicated linguistic properties of hypernymy
  – Extending BiRRE to address other similar tasks, such as graded lexical entailment
  – Exploring how deep neural language models can improve the performance of hypernymy detection
Thank You!

Questions & Answers