Association for Computational Linguistics

华东师范大学计算机科学与技术学院
School of Computer Science and Technology

BiRRE: Learning Bidirectional Residual Relation Embeddings for Supervised Hypernymy Detection

Chengyu Wang[1,2], Xiaofeng He[3*]

[1] School of Software Engineering, East China Normal University, Shanghai, China

[2] Alibaba Group, Hangzhou, China

[3] School of Computer Science and Engineering, East China Normal University, Shanghai, China

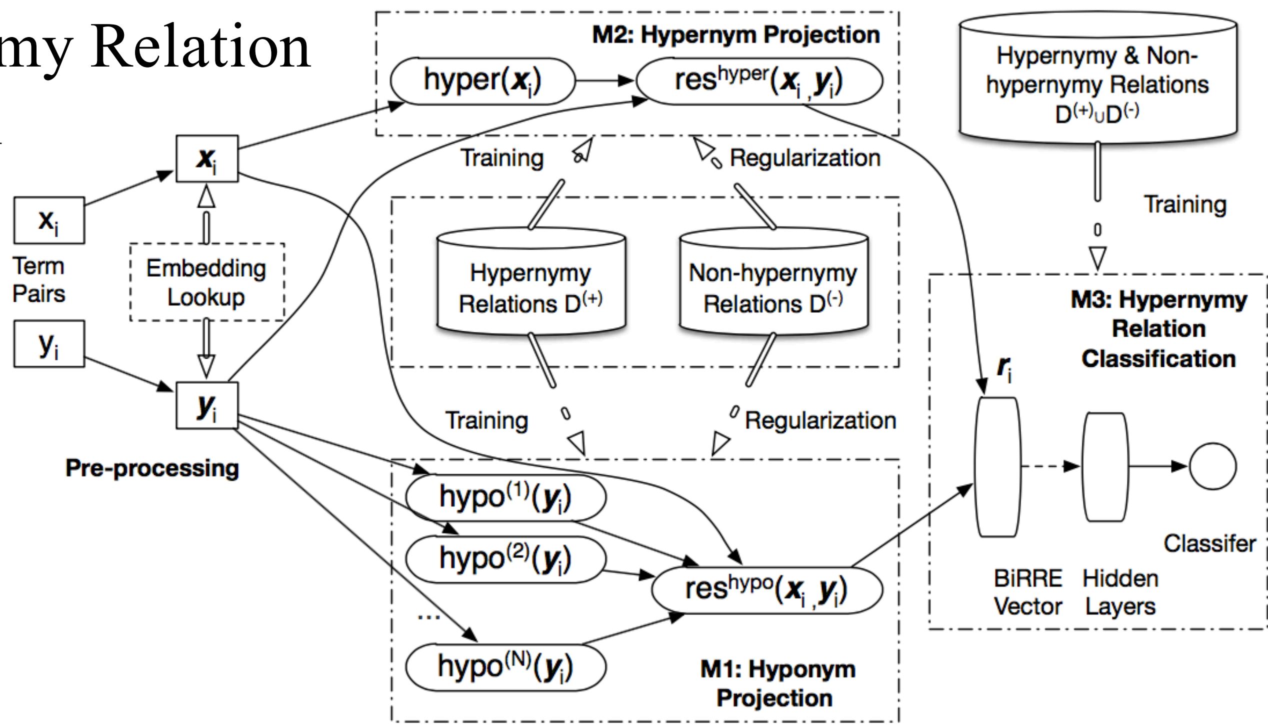
软件工程学院
SOFTWARE ENGINEERING INSTITUTE

Alibaba Group
阿里巴巴集团

Outline

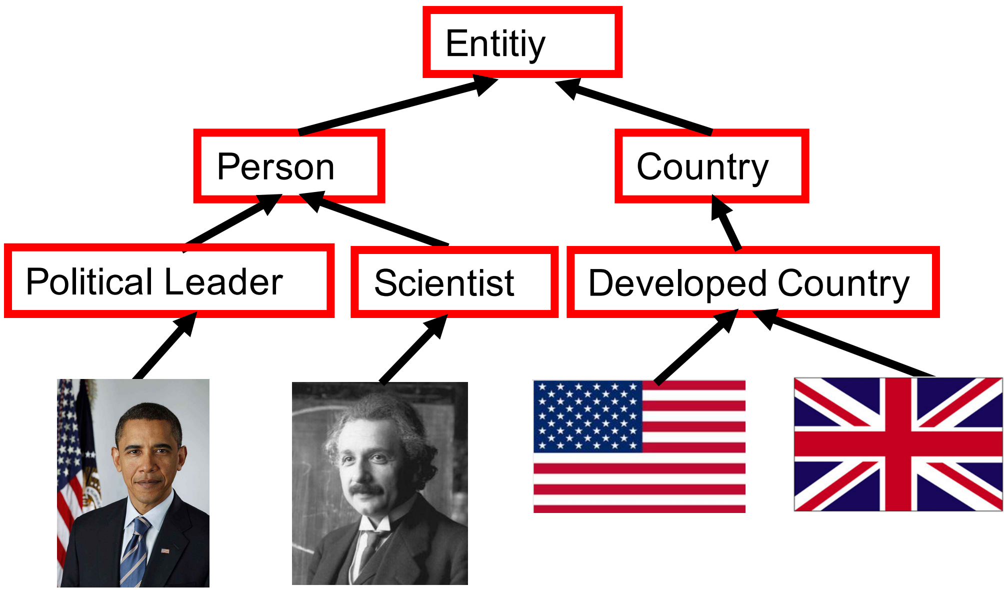
- Introduction
- The BiRRE Model
  - M1: Hyponym Projection
  - M2: Hypernym Projection
  - M3: Hypernymy Relation Classification
- Experiments
- Conclusion

General Framework



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



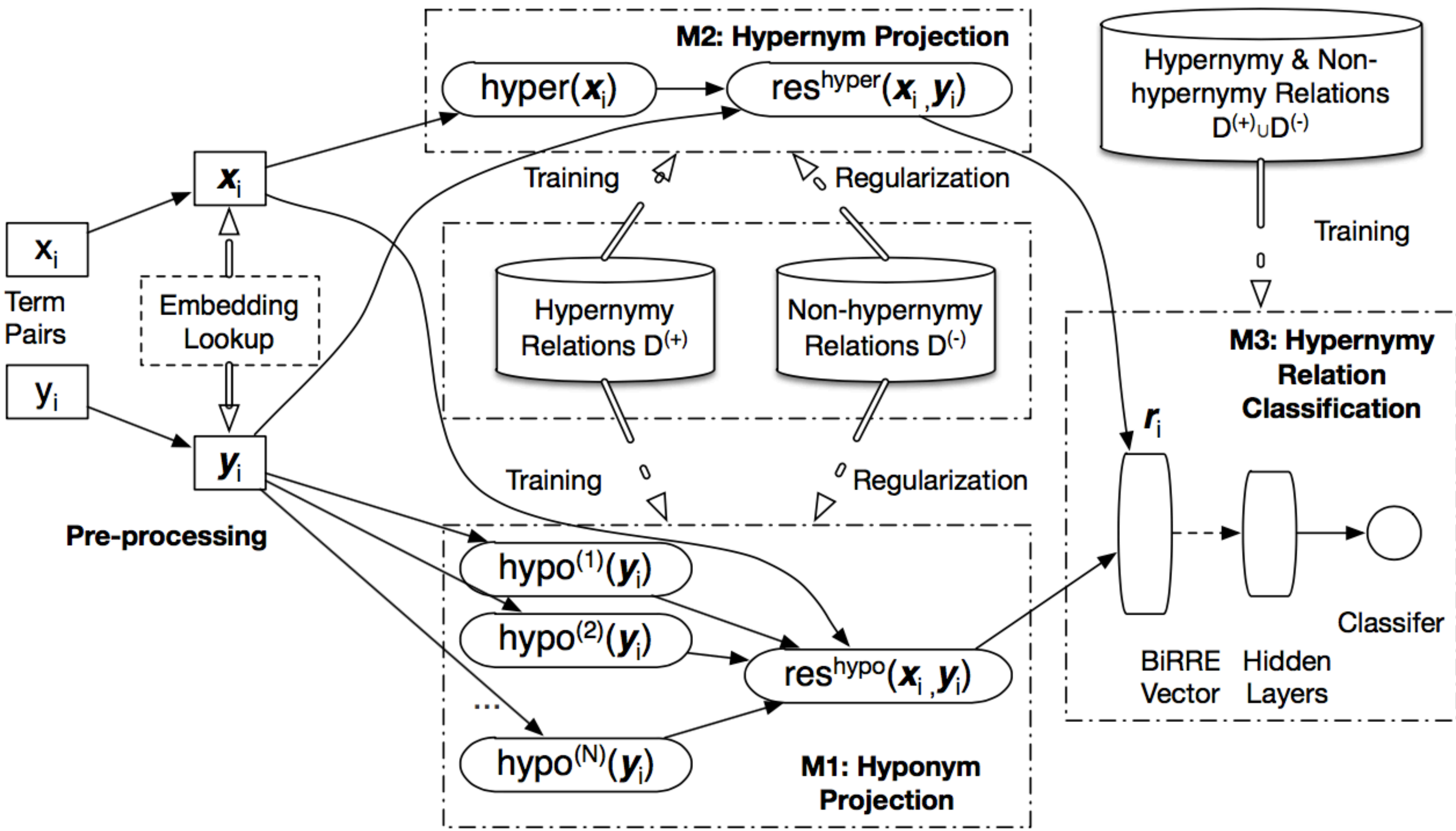
Introduction (2)

- 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
    - $\vec{x}$ can be projected to $\vec{y}$ by hypernym projection
    - $\vec{y}$ can be projected to $\vec{x}$ by hyponym projection

Positive sample:
(cat ,mammal)

Negative sample:
(desk, fruit)

BiRRE: The Proposed Framework

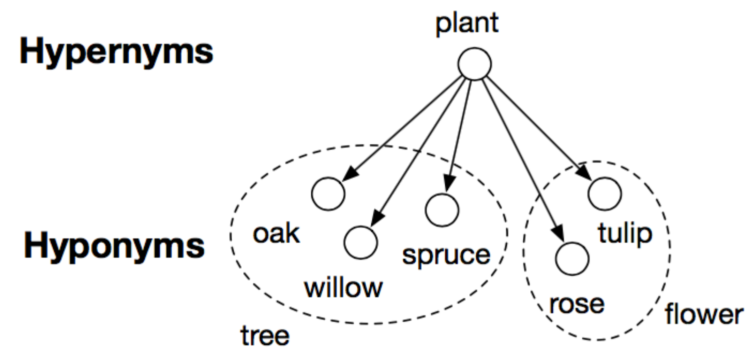


Hyponym Projection (M1)

- Learning N projection matrices from hypernyms to hyponyms
  - Simple objective function

$$\min_{\mathcal{M}} \sum_{(x_i,y_i)\in D^{(+)}} \sum_{p=1}^{N} \theta_i^{(p)} \|\mathbf{M}^{(p)}\mathbf{y}_i - \mathbf{x}_i\|^2$$

$$\text{s. t.} \quad {\mathbf{M}^{(p)}}^T \mathbf{M}^{(p)} = \mathbf{I}_d, p \in \{1, \cdots, N\}$$



(a) Hypernym–to–hyponym mappings

  - Considering negative regularization

$$\min_{\mathcal{M}} \frac{1}{|D^{(+)}|} \sum_{(x_i,y_i)\in D^{(+)}} \sum_{p=1}^{N} \theta_i^{(p)} \|\mathbf{M}^{(p)}\mathbf{y}_i - \mathbf{x}_i\|^2$$

$$+ \frac{\lambda}{|D^{(-)}|} \sum_{(x_i,y_i)\in D^{(-)}} \sum_{p=1}^{N} \phi_i^{(p)} (\mathbf{M}^{(p)}\mathbf{y}_i)^T \cdot \mathbf{x}_i$$

$$\text{s. t.} \quad {\mathbf{M}^{(p)}}^T \mathbf{M}^{(p)} = \mathbf{I}_d, p \in \{1, \cdots, N\}$$

No standard off-the-shelf learning algorithm!

Hyponym Projection (M1)

- Efficient learning algorithm for hyponym projection
  - Slight changes of the objective function

$$\min_{\mathcal{M}} \frac{1}{|D^{(+)}|} \sum_{(x_i,y_i)\in D^{(+)}} \sum_{p=1}^{N} \theta_i^{(p)} \|\mathbf{M}^{(p)}\mathbf{y}_i - \mathbf{x}_i\|^2$$

$$- \frac{\lambda}{|D^{(-)}|} \sum_{(x_i,y_i)\in D^{(-)}} \sum_{p=1}^{N} \phi_i^{(p)} \|\mathbf{M}^{(p)}\mathbf{y}_i - \mathbf{x}_i\|^2$$

$$\text{s. t.} \quad {\mathbf{M}^{(p)}}^T \mathbf{M}^{(p)} = \mathbf{I}_d, p \in \{1, \cdots, N\}$$

Latent Projection Model with Negative Regularization (LPMNR)

  - Learning projection matrices

1: **for** $p = 1$ to N **do**
2: $\mathbf{B}^{(p)} = \sum_{(x_i,y_i)\in D^{(+)}} \theta_i^{(p)} \mathbf{x}_i \mathbf{y}_i^T$ $-\alpha \cdot \sum_{(x_i,y_i)\in D^{(-)}} \phi_i^{(p)} \mathbf{x}_i \mathbf{y}_i^T$;
3: $\mathbf{U}^{(p)}\mathbf{\Sigma}^{(p)}\mathbf{V}^{(p)T} = SVD(\mathbf{B}^{(p)})$;
4: $\mathbf{R}^{(p)} = \text{diag}(\underbrace{1, \ldots, 1}_{d-1}, \det(\mathbf{U}^{(p)})\det(\mathbf{V}^{(p)}))$;
5: $\mathbf{M}^{(p)} = \mathbf{U}^{(p)}\mathbf{R}^{(p)}\mathbf{V}^{(p)T}$;
6: **end for**

Iterative Optimization

- Learning latent variables

$$\theta_i^{(p)*} = \theta_i^{(p)} - \eta \cdot \sum_{(x_i,y_i)\in D^{(+)}} \|\mathbf{M}^{(p)}\mathbf{y}_i - \mathbf{x}_i\|^2$$

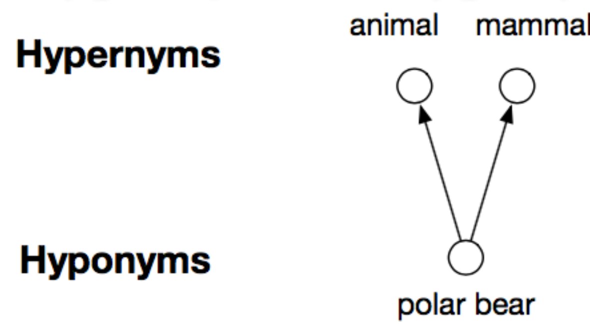
$$\phi_i^{(p)*} = \phi_i^{(p)} + \eta \cdot \sum_{(x_i,y_i)\in D^{(-)}} \|\mathbf{M}^{(p)}\mathbf{y}_i - \mathbf{x}_i\|^2$$

* 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

$$\min_{\mathbf{Q}} \frac{1}{|D^{(-)}|} \sum_{(x_i,y_i)\in D^{(-)}} \|\mathbf{Q}\mathbf{x}_i - \mathbf{y}_i\|^2 - \frac{\lambda}{|D^{(+)}|} \sum_{(x_i,y_i)\in D^{(+)}} \|\mathbf{Q}\mathbf{x}_i - \mathbf{y}_i\|^2 \ \text{s. t.} \ \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_d$$



(b) Hypnoym–to–hypernym mappings

  - Learning algorithm: a simpler version of M1
- Training of the hypernymy relation classifier
  - Hyponym residual vector: $res^{hypo}(\mathbf{x}_i, \mathbf{y}_i)) = \mathbf{x}_i - \mathbf{M}^{(\tilde{p})}\mathbf{y}_i$
  - Hypernym residual vector: $res^{hyper}(\mathbf{x}_i) = \mathbf{Q}\mathbf{x}_i - \mathbf{y}_i$
  - Feature representations: $\mathbf{r}_i = res^{hypo}(\mathbf{x}_i, \mathbf{y}_i) \oplus res^{hyper}(\mathbf{x}_i, \mathbf{y}_i)$
  - Classifier learning: simple back propagation training of feed-forward neural networks

Experiments (1)

- Experimental Settings
  - Word embeddings: fastText embeddings, $d = 300$
  - Default parameters settings:
    - $\eta = 0.001, N = \max\{1, \lfloor \lg |D^{(+)}| \rfloor\}$
  - Optimization: Adam with dropout rate 0.1
- Effectiveness of BiRRE over the largest dataset (Shwartz et al. 2016)

Method	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
	Random Split			Lexical Split		
Roller and Erk (2016)	0.926	0.850	0.886	0.700	**0.964**	0.811
Shwartz et al. (2016)	0.913	0.890	0.901	0.809	0.617	0.700
Glavas and Ponzetto (2017)	0.933	0.826	0.876	0.705	0.785	0.743
Rei et al. (2018)	0.928	0.887	0.907	0.826	0.860	0.842
BiRRE	**0.945**	**0.932**	**0.938**	**0.880**	0.918	**0.898**

Experiments (2)

- General Performance
  - Results over two general benchmark datasets
    - BLESS
    - ENTAILMENT
- Ablation Study
  - Choice of baselines
    - Addition, offset and concatenation of term vectors
    - Unidirectional residual vectors

Method	BLESS	ENT.
Mikolov et al. (2013)	0.84	0.83
Yu et al. (2015)	0.90	0.87
Luu et al. (2016)	0.93	0.91
Nguyen et al. (2017)	0.94	0.91
Wang et al. (2019a)	0.97	0.92
BiRRE	**0.98**	**0.93**

Feature Set	BLESS	ENT.	Shwartz
$\mathbf{x}_i + \mathbf{y}_i$	0.76	0.77	0.72
$\mathbf{x}_i - \mathbf{y}_i$	0.79	0.74	0.73
$\mathbf{x}_i \oplus \mathbf{y}_i$	0.81	0.80	0.77
$res^{hypo}(\mathbf{x}_i, \mathbf{y}_i)$	0.92	0.87	0.84
$res^{hyper}(\mathbf{x}_i, \mathbf{y}_i)$	0.89	0.84	0.82
$\mathbf{r}_i$ (i.e., BiRRE)	**0.99**	**0.93**	**0.88**

* 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