Embedding Entities and Relations for Learning and Inference in Knowledge Bases

Authors: Bishan Yang et al.
Presented by: Aarash

Overview

- Motivations
- Previous Methods
- General Framework
- Simplified Model
- Link Prediction
- Rule Extraction
- Conclusion

Motivations

- Finding a unified learning framework and presenting a new method
  - E.g. Translation Embedding (TransE), Neural Tensor Network (NTN), etc.
- Link Prediction
  - Gained 73.2% vs. 54.7% by TransE on FreeBase
- Mining logical rules
  - E.g. *BornInCity(a, b) ∧ CityInCountry(b, c) ⇒ Nationality(a, c)*

Previous Methods

Triplet in KB: (e_1, r, e_2)

- NTN
  - Represents relations as **bilinear tensor operator** followed by a **linear matrix operator**
  - Represents entities as average of word vectors (initialized with pre-trained vectors)
- TransE
  - Represents relations as a **single vectors**
  - Represents entities as unit vectors (one-hot encoding)

General Framework

- Entity representations

$$\mathbf{y}_{e_1} = f\left(\mathbf{W}\mathbf{x}_{e_1}\right), \;\; \mathbf{y}_{e_2} = f\left(\mathbf{W}\mathbf{x}_{e_2}\right)$$

- Relation representations
  - Linear Transformation

$$g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{A}_r^T \begin{pmatrix} \mathbf{y}_{e_1} \\ \mathbf{y}_{e_2} \end{pmatrix}$$

  - Bilinear Transformation

$$g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{y}_{e_1}^T \mathbf{B}_r \mathbf{y}_{e_2}$$

General Framework (cont.)

Table 1. Comparisons among several multi-relational models in their scoring functions

Models	$\mathbf{B}_r$	$\mathbf{A}_r^T$	Scoring Function
Distance (Bordes et al., 2011)	-	$\begin{pmatrix}\mathbf{Q}_{r_1}^T & -\mathbf{Q}_{r_2}^T\end{pmatrix}$	$-\lvert\lvert g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2})\rvert\rvert_1$
Single Layer (Socher et al., 2013)	-	$\begin{pmatrix}\mathbf{Q}_{r1}^T & \mathbf{Q}_{r2}^T\end{pmatrix}$	$\mathbf{u}_r^T \tanh(g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}))$
TransE (Bordes et al., 2013b)	$\mathbf{I}$	$\begin{pmatrix}\mathbf{V}_r^T & -\mathbf{V}_r^T\end{pmatrix}$	$-(2g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) - 2g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) + \lvert\lvert\mathbf{V}_r\rvert\rvert_2^2)$
NTN (Socher et al., 2013)	$\mathbf{T}_r$	$\begin{pmatrix}\mathbf{Q}_{r1}^T & \mathbf{Q}_{r2}^T\end{pmatrix}$	$\mathbf{u}_r^T \tanh\left(g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) + g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2})\right)$

$$\mathbf{y}_{e_1}, \mathbf{y}_{e_2} \in R^n$$

$$\mathbf{Q}_{r_1}, \mathbf{Q}_{r_2} \in R^{n \times m}$$

$$\mathbf{T}_r \in R^{n \times n \times m}$$

$$\mathbf{V}_r \in R^n$$

Simplified Model

- Bilinear model:

$$g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{y}_{e_1}^T \mathbf{M}_r \mathbf{y}_{e_2} \qquad \mathbf{M}_r \in R^{n \times n}$$

- Bilinear-diag model (simple presented model):
  - $\mathbf{M}_r$ is a diagonal matrix.
  - Same number of parameters as TransE
  - Loss function:

$$L(\Omega) = \sum_{(e_1,r,e_2)\in T} \sum_{(e_1',r,e_2')\in T'} \max\{S_{(e_1',r,e_2')} - S_{(e_1,r,e_2)} + 1, 0\}$$

Link Prediction

- Datasets
  - FreeBase (FB14K)
  - WordNet (WN)
- Models
  - NTN with 4 tensor slices
  - Bilinear+Linear NTN with 1 tensor slice without non-linear layer
  - TransE, special case of Bilinear+Linear (DistAdd)
  - Bilinear and Bilinear-diag (DistMult)
- Evaluations
  - Mean Reciprocal Rank (MRR)
  - HITS@10
  - Mean Average Precision (MAP)

Link Prediction (cont.)

Table 2. Performance comparisons among different embedding models

	FB15k		FB15k-401		WN	
	MRR	HITS@10	MRR	HITS@10	MRR	HITS@10
NTN	0.25	41.4	0.24	40.5	0.53	66.1
Blinear+Linear	0.30	49.0	0.30	49.4	0.87	91.6
TransE (DISTADD)	0.32	53.9	0.32	54.7	0.38	90.9
Bilinear	0.31	51.9	0.32	52.2	**0.89**	92.8
Bilinear-diag (DISTMULT)	**0.35**	**57.7**	**0.36**	**58.5**	0.83	**94.2**

- Performance decreases as complexity increases (due to overfitting)

Link Prediction (cont.)

- DistAdd:
  - Relations between entities based on additions
  - If (a, r, b) ⇒ $y_a + V_r \simeq y_b$ (where V_r is a vector)

- DistMult:
  - Relations between entities based on multiplications
  - If (a, r, b) ⇒ $y_a^T M_r \simeq y_b^T$ (where M_r is a diagonal matrix)

Link Prediction (cont.)

- Models based on basic DistMult:
    - DistMult: Bilinear-diag
    - DistMult-tanh: using *tanh* for entity projection
    - DistMult-tanh-EV-init: Initializing 1000d pre-trained entity vectors
    - DistMult-tanh-WV-init: Average of the 300d word vectors in each entity

Table 3. Evaluation with pretrained vectors on FB15K-401

	MRR	HITS@10	MAP (w/ type checking)
DISTMULT	0.36	58.5	64.5
DISTMULT-tanh	0.39	63.3	76.0
DISTMULT-tanh-WV-init	0.28	52.5	65.5
DISTMULT-tanh-EV-init	**0.42**	**73.2**	**88.2**

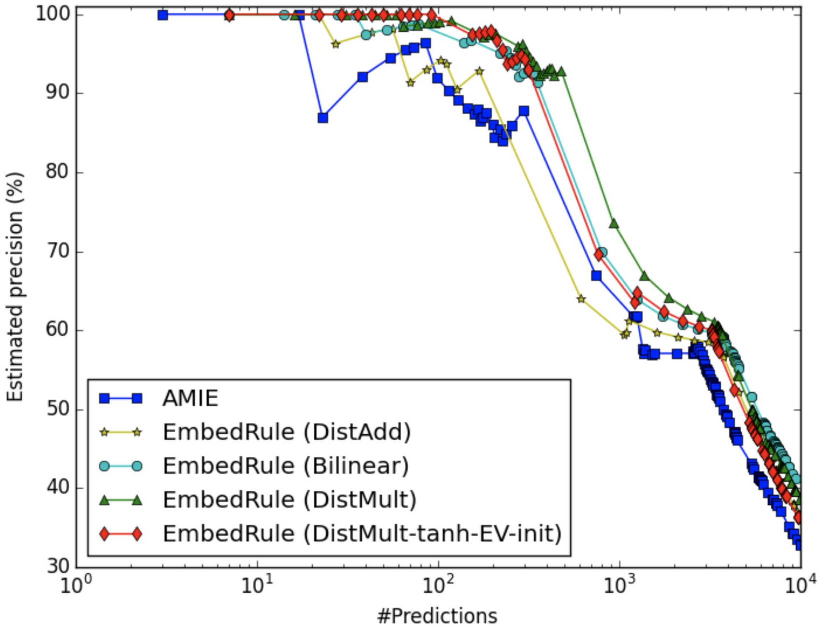
Rule Extraction

- Motivations
  - Scalable to large KBs
  - More generalizable method for rule extraction
- As multiplications or additions of two relation embeddings

$$\begin{aligned}\mathbf{y}_a + \mathbf{V}_1 &\approx \mathbf{y}_b \\ \mathbf{y}_b + \mathbf{V}_2 &\approx \mathbf{y}_c\end{aligned} \Longrightarrow \mathbf{y}_a + (\mathbf{V}_1 + \mathbf{V}_2) \approx \mathbf{y}_c$$

$$\begin{aligned}\mathbf{y}_a^T \mathbf{M}_1 &\approx \mathbf{y}_b^T \\ \mathbf{y}_b^T \mathbf{M}_2 &\approx \mathbf{y}_c^T\end{aligned} \Longrightarrow \mathbf{y}_a^T (\mathbf{M}_1 \mathbf{M}_2) \approx \mathbf{y}_c^T$$

Rule Extraction (cont.)



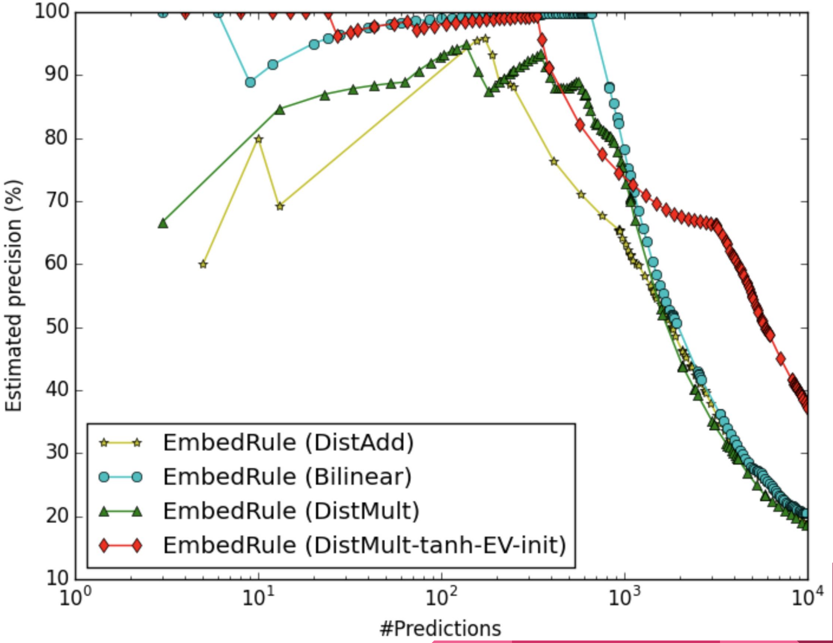


Figure 1. Aggregated precision on length-2 (left) and length-3 (right) rules extracted by different methods

Conclusion

- Limitations
  - Bilinear-Diag trouble encoding difference between a relation and its inverse
  - Incomplete explanation for rule extraction observations
  - No results on WordNet
- Future directions
  - Deep structures for neural network framework
  - Capturing hierarchical structure hidden in the multi-relational data
  - Tensors constructs and architectures may improve multi-relational learning

Thank you!

Any questions?