CSCI 669

Embedding entities and relations for learning and interence in Knowledge bases

Presenter: Hayley Song

Date: 11-15-2018

Topic:

- learning representations of entities and relations in KBs using the neural-embedding approach

Three contributions

1. Presents a general framework for multi-relational learning that unifies most multi-relational embedding models (eg. NTN, TransE)
    - eneities = low-dim vectors learned from a NN
    - relations = bilinear and/or linear mapping
2. Empirically evaluates different choices of entity representations and relation representations under this framework on link prediction task
    - shows a simple bilinear formulation achieves new state-of-the-art
3. Introduce a **"embedding-based rule extraction"**: new approach to mine logical rules (eg. BornInCity(a,b) ^ CityInCountry(b,c) -> Nationality(a,c)) using the learned relation embeddings
    - shows that such rules can be effectively extracted by modeling the composition of relation embeddings
    - embedding learned from the bilinear objective captures well compositional semantics or relaions via matrix multiplication
    - outperforms AMIE on mining rules that involve compositional reasoning

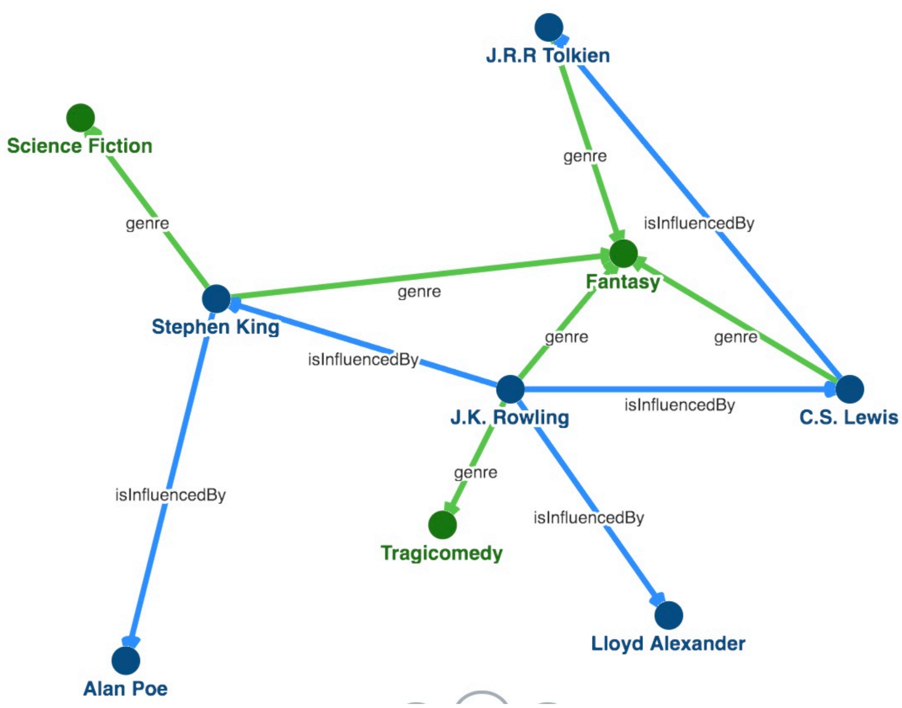
Lesson learned

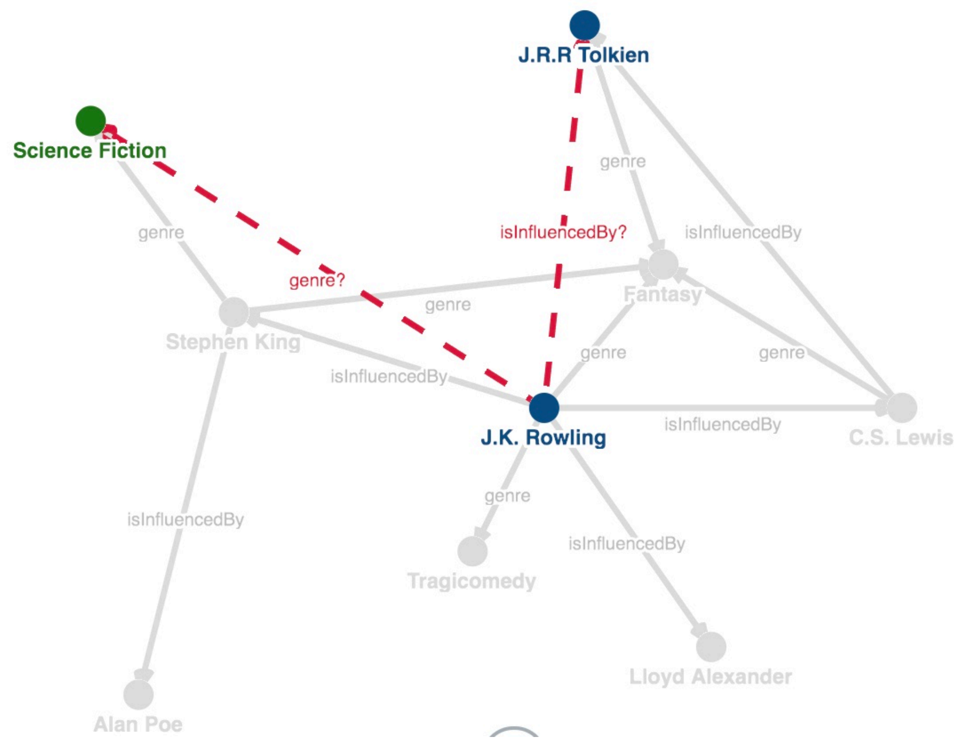
1. embedding learned from the bilinear objective are good at capturing relational semantics
2. composition of relations is characterized by matrix multiplication
3. their new embedding-based rule extraction approach achieves the state-of-the-art in mining Horn rules that involve compositional reasoning

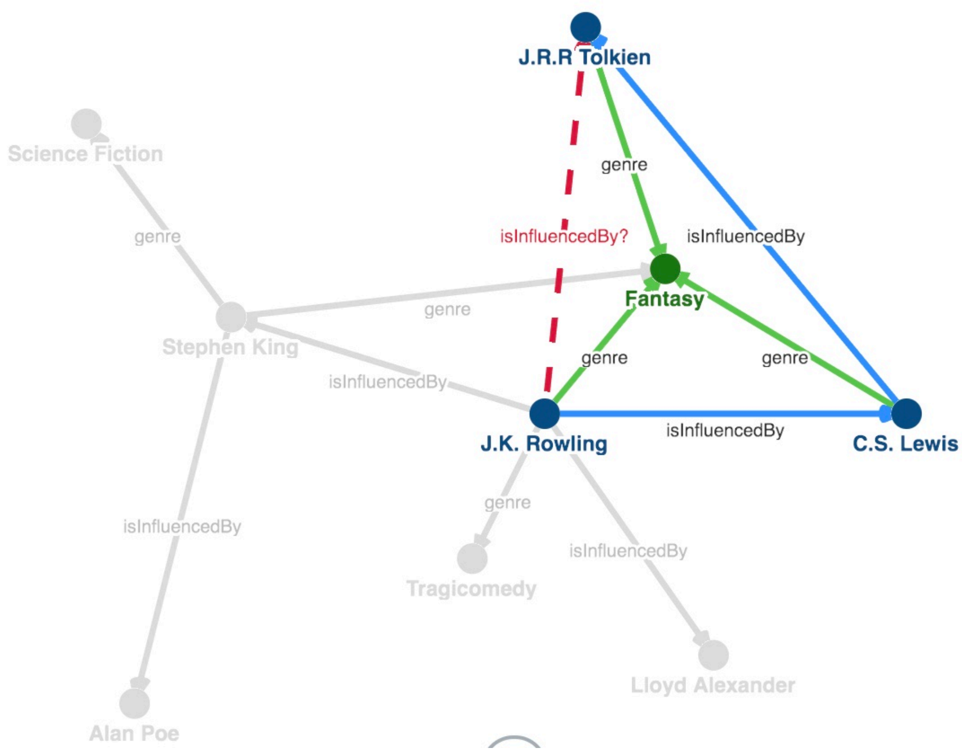
Set up

Large Knowledge bases: Freebase, DBPedia, YAGO

- entities and relations in RDF triple form (eg. (sub, pred, obj))
- really large: millions of entities, various relaions and billions of triples
- Why are these KBs interesting?
    - they can be used to improve various tasks, eg. information retrieval, QA, biological data mining
    - "Relational learning": learning the relations between entities from these large knowledge bases. Use low-dim representations of entities and relations
        i. Tensor factorization
        ii. Neural-embedding-based models [focus]
        - representations of entities and relations are learned using NN
          Both have shown good scalability and reasoning (in terms of validating unseen facts given an existing KB) (aka. generalizability)
- Link prediction in KBs (aka. KG completion)
    a. KBs are incomplete: missing some entities and relations
    b. learn from local and global connectivity pattern from entities and relationships of different types at the same time
    c. Relation predicitions are then performed by using the learned patterns to generalize observed relationships between an entity of interest and all others







Related work

Markov-logic networks

- traditional statistical learning approach
- does not scale well

Different approach: embed multi-relational knowledge into low-dim representations of entities and relations

- improved scalability
- strong generalizability/reasoning on large-scale KBs
  Consider a training set S of triplets (h, l, t) where $h, t \in E$ (set of entities) and $l \in L$ set of relations.

1. TransE
   Relationships as translations in the embedding space (Bordes et al., 2013 NIPS)

- **Key**
  If (h, l, t)holds, then the embedding of the t should be close to the h plus some vector that depends on the relationship (l). Otherwise, $h + l$ should be far away from t

- **Learning**
  Minimize a margin-based ranking criterion over the training set with norm-constraints on entity vectors: the L2-norm of the embeddings of the entities is 1
  $(\| \boldsymbol{h} \|_2^2 = \| \boldsymbol{t} \|_2^2 = 1)$

  - This prevents the training process to trivially minimize L by increating entity embedding norms proportionally

  - corrupted triplets: training triplets with either the head or tail replaced by a random entitiy (but not both at the same time)

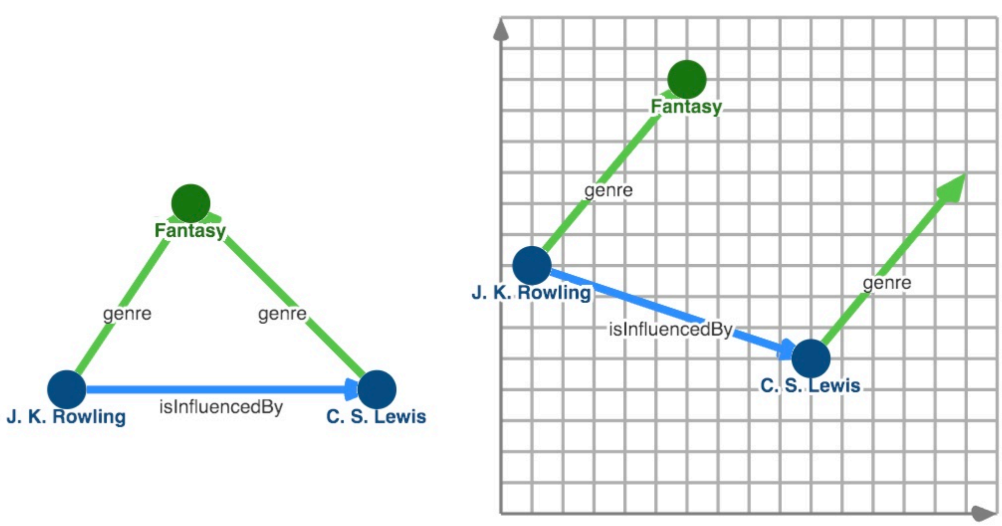
  - margin-based loss:

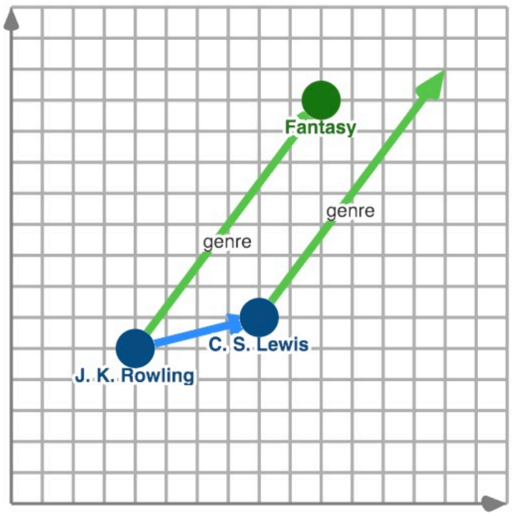
$$\mathcal{L} = \sum_{(h,\ell,t) \in S} \sum_{(h',\ell,t') \in S'_{(h,\ell,t)}} \left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+$$

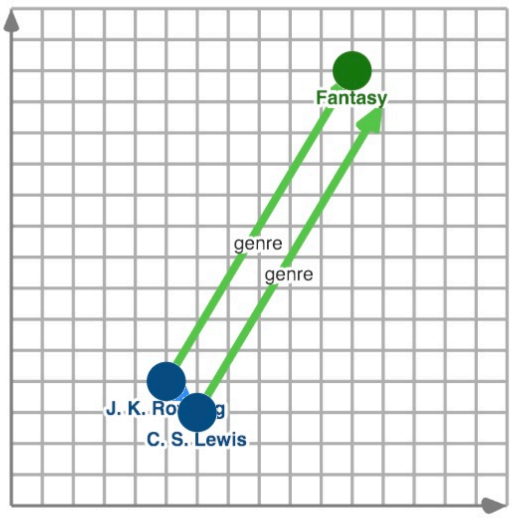
where $[x]_+$ denotes the positive part of x, $\gamma > 0$ is a margin hyperparameter, and

$$S'_{(h,\ell,t)} = \left\{(h', \ell, t) | h' \in E\right\} \cup \left\{(h, \ell, t') | t' \in E\right\}.$$

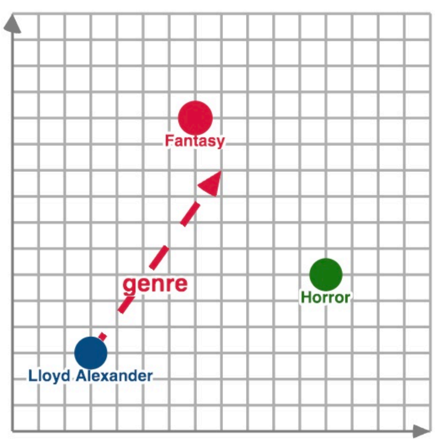
- **Training demo**





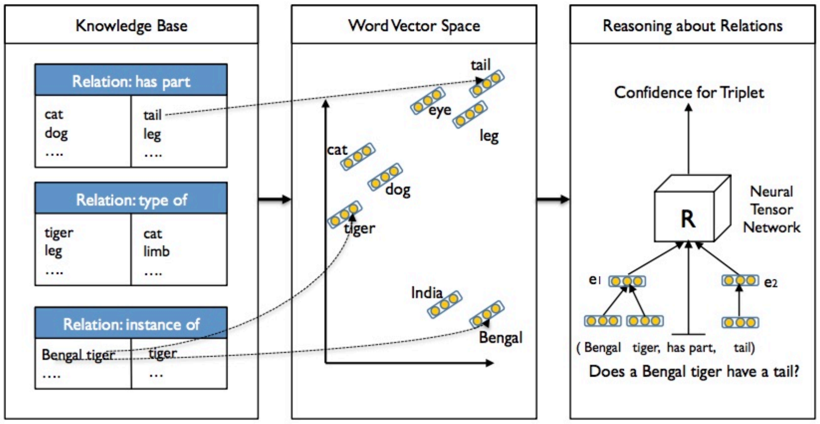


- **Prediction**
  Q: (Loyd Alexander, genre, ?), what is Loyd Alexander's genre?
  - Add the translation vector of "genre" relationship with "Loyd Alexander" entity vector and choose the nearest entity



[Image credit]()

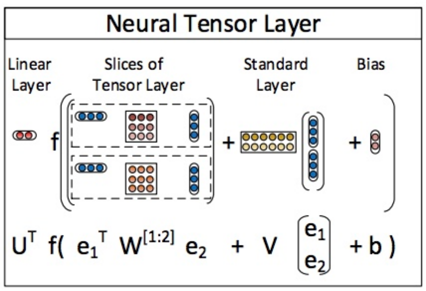
2. [Neural Tensor Network (NTN)]()
   Goal: state whether two entities (h, t) are in a certain relationship l and with what certainty
   eg) (h, l, t) = (Bental tiger, has part, tail) is true? with what certainty?
   Overview



Key

Replace a standard linear NN layer with a bilinear tensor layer that relates the two entity vectors across multiple dimensions: "Neural tensor layer"

eg:) tensor layer with two slices



Math says:

$$g(e_1, R, e_2) = u_R^T f\left(e_1^T W_R^{[1:k]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R\right), \tag{1}$$

where $f = \tanh$ is a standard nonlinearity applied element-wise, $W_R^{[1:k]} \in \mathbb{R}^{d \times d \times k}$ is a tensor and the bilinear tensor product $e_1^T W_R^{[1:k]} e_2$ results in a vector $h \in \mathbb{R}^k$, where each entry is computed by one slice $i = 1, \ldots, k$ of the tensor: $h_i = e_1^T W_R^{[i]} e_2$. The other parameters for relation R are the standard form of a neural network: $V_R \in \mathbb{R}^{k \times 2d}$ and $U \in \mathbb{R}^k, b_R \in \mathbb{R}^k$.

vs. transE

$$g(e_1, R, e_2) = \|W_{R,1}e_1 - W_{R,2}e_2\|_1$$

Score function

$$s(h, \ell, t) = \boldsymbol{h}^T \boldsymbol{L} \boldsymbol{t} + \boldsymbol{\ell}_1^T \boldsymbol{h} + \boldsymbol{\ell}_2^T \boldsymbol{t}$$

where $\boldsymbol{L} \in \mathbb{R}^{k \times k}$, $\boldsymbol{L}_1 \in \mathbb{R}^k$ and $\boldsymbol{L}_2 \in \mathbb{R}^k$, all of them depending on ℓ.

Training Objective

Same as in TransE (contrasitve max-margin)

$$J(\boldsymbol{\Omega}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \max\left(0, 1 - g\left(T^{(i)}\right) + g\left(T_c^{(i)}\right)\right) + \lambda \|\boldsymbol{\Omega}\|_2^2,$$

Two improvements

a. represent entities by the average of its word vectors: *compositionality of language*
   eg:) vec(homo-sapiens) = vec("hominid") + vec("sapiens")
   This can help with generalizability for predicting relations on unseen entities, eg. *homo-erectus*
b. Initilize the word vectors with pre-trained vectors
   This takes advantage of general syntactic and semantic information from larger corpus.

Images credit: original paper

This paper

Notation:

- Triplet: (e_1, R, e_2) where e_1 is the head entity (subject), R is the relation and e_2 is the tail (object)
- $y_e \in \mathcal{R}^n$: entity representation of entity e in n-dimsion
- Q_r (or V_r): $n \times m$ matrix (or m-dim vector) for linear transformation g_r^a
- $T_r \in \mathcal{R}^{n \times n \times m}$: bilinear transformation for g_r^b

Contribution1: General NN framework for multi-relational representation learning

1. Entity representations: project the input entity (x_{e1}, x_{e2}) to (y_{e1}, y_{e2}):

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

   where f can be a linear or non-linear function, and $\mathbf{W}$ is a parameter matrix, which can be randomly initialized or initialized using pre-trained vectors.

2. Relation representations
   Most existing scoring functions can be unified based on a basic linear transformation g_r^a, a bilinear transformation g_r^b, or their combination, where g_r^a are g_r^b are defined as :

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

   which $\mathbf{A}_r$ and $\mathbf{B}_r$ are relation-specific parameters.

3. Summary of reformulated popular scoring functions

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

   Note:

   - NTN is the most expressive model: contains both linear and bilinear relation operators
   - TransE is the simplest model (fewest params): only linear relation operators with one-dim vectors
   - Basic bilinear scoring function: $g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{y}_{e_1}^T \mathbf{M}_r \mathbf{y}_{e_2}$

- special case of NTN w/o non-libnear layer and the linear operators, uses 2-d matrix operator $M_r \in \mathcal{R}^{n \times n}$ rather than a tensor operator
- Further constraint: M_r is diagonal

4. Training objective
   Minimize the margin-based ranking loss:

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

Experiment 1

Link prediction: Predict the correctness of unseen triplets

- Dataset: WordNet (WN), Freebase (FB15k)
- Metrics: Mean Reciprocal Rank(MRR), HITS@10 (top-10 accuracty), Mean Average Precision (MAP)
- Models
  a. NTN with 4 tensor slices as in (Socher et al., 2013)
  b. Bilinear+Linear, NTN with 1 tensor slice and without the non-linear layer
  c. TransE, a special case of Bilinear+Linear
  d. Bilinear: using scoring function above
  e. Bilinear-diag: a special case of Bilinear where the relation matrix is a diagonal matrix
- Result

	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**

Table 2: Performance comparisons among different embedding models

  - BILINEAR performs better than TransE, esp. on WN: captures more expressive relations
  - Multiplicative vs. Addivtive interactions
    Overall superior performance of BILINEAR-DIAG (*DISTMULT*)than TransE(*DISTADD*)

	Predicting subject entities				Predicting object entities			
	1-to-1	1-to-n	n-to-1	n-to-n	1-to-1	1-to-n	n-to-1	n-to-n
DISTADD	70.0	76.7	21.1	53.9	68.7	17.4	**83.2**	57.5
DISTMULT	**75.5**	**85.1**	**42.9**	**55.2**	**73.7**	**46.7**	81.0	**58.8**

Table 3: Results by relation categories: one-to-one, one-to-many, many-to-one and many-to-many

Contribution2: Embedding-based rule extraction

Use the learned embedding to extract logical rules from the KB

- Key is how to effectively explore the search space
- Proposed method's efficiency is affected by the number of distinct relation types (usually relatively small), not by the size of the KB graph
- Limit: Restricted to Horn rules, eg:
  $BornInCity(a, b) \wedge CityOfCountry(b, c) \implies Nationality(a, c)$
- Consider Horn rule of length 2
  $B_1(a, b) \wedge B_2(b, c) \implies H(a, c)$

  a. Body of the rule can be viewed as the composition of relations B_1 and B_2, a new relation that has the property that entities a and c are in a relation iff there is an entity b which satifies B_1 and B_2

  b. Model relation composition as multiplication(or addition) of two relation embeddings

     - addition for relation vector embeddings
       If $y_a + V \approx y_b$ when $B(a, b)$ holds, then $y_a + V_1 \approx y_b$ and $y_b + V2 \approx y_c$ implies $y_a + (V_1 + V_2) \approx y_c$
     - multiplication for relation matrix embeddings
       Given ys are unit vectors and M is the bilinear transformation, when $B(a, b)$ holds if $y_a^T M y_b \approx 1$ and $y_a^T M$ is still a unit vector, then $y_a^T M_1 \approx y_b^T$ and $y_a^T M_2 \approx y_b^T 1$ implies $y_a^T (M_1 M_2) \approx y_c^T$

  c. Similarity score
     The composition of B_1 and B_2 should be close to H in L_2 (if vector) or Frobenius norm (if matrix)
     This distance metric allows us to rank possible pairs of relations with respect to the relevance of their composition to the target relation

  d. Reduce the search domain by argument (entities) type constraints

     - If the relation in the head is r, then we are only interested in relation pairs (p, q) s.t. (1) $\mathcal{Y}_p \cap \mathcal{X}_q \neq \emptyset$; (2) $\mathcal{X}_p \cap \mathcal{X}_r \neq \emptyset$; (3) $\mathcal{Y}_q \cap \mathcal{Y}_r \neq \emptyset$

- Algorithm

Algorithm 1 EMBEDRULE

```
Input: KB = {(e_1, r, e_2)}, relation set R
Output: Candidate rules Q
for each r in R do
    Select the set of start relations S = {s : X_s ∩ X_r ≠ ∅}
    Select the set of end relations T = {t : Y_t ∩ Y_r ≠ ∅}
    Find all possible relation sequences
    Select the K-NN sequences P' ⊆ P for r based on dist(M_r, M_{p_1} ∘ · · · ∘ M_{p_n})
    Form candidate rules using P' where r is the head relation and p ∈ P' is the body in a rule
    Add the candidate rules into Q
end for
```

- Examples of extracted Horn rules

A EXAMPLES OF THE EXTRACTED HORN RULES

Examples of length-2 rules extracted by EMBEDRULE with embeddings learned from DISTMULT-tanh-EV-init:

$$AwardInCeremany(a,b) \wedge CeremanyEventType(b,c) \implies AwardInEventType(a,c)$$
$$AtheletePlayInTeam(a,b) \wedge TeamPlaySport(b,c) \implies AtheletePlaySport(a,c)$$
$$TVProgramInTVNetwork(a,b) \wedge TVNetworkServiceLanguage(b,c) \implies TVProgramLanguage(a,c)$$
$$LocationInState(a,b) \wedge StateInCountry(b,c) \implies LocationInCountry(a,c)$$
$$BornInLocation(a,b) \wedge LocationInCountry(b,c) \implies Nationality(a,c)$$

Examples of length-3 rules extracted by EMBEDRULE with embeddings learned from DISTMULT-tanh-EV-init:

$$SportPlayByTeam(a,b) \wedge TeamInClub(b,c) \wedge ClubHasPlayer(c,d) \implies SportPlayByAthelete(a,d)$$
$$MusicTrackPerformer(a,b) \wedge PeerInfluence(b,c) \wedge PerformRole(c,d) \implies MusicTrackRole(a,d)$$
$$FilmHasActor(a,b) \wedge CelebrityFriendship(b,c) \wedge PersonLanguage(c,d) \implies FilmLanguage(a,d)$$

Experiment 2. Rule extraction

1. Dataset: FB15k-401 (485,741 facts, 14,417 entities, and 373 relations)

2. Metric

    - Precision = ratio of predictions that are in the test (unseen) data to all the generted unseen predictions

3. Models
   Embeddings trained from

    - TRANSE (DISTADD)
    - BILINEAR
    - BILINEAR-DIAG (DISTMULT)
    - DISTMULT-tanh-EV-init

4. Results

Figure 1 compares the predictions generated by the length-2 rules extracted by different methods. From left to right, the n-th data point represents the total number of predictions of the top
n rules and the estimated precision of these predictions.

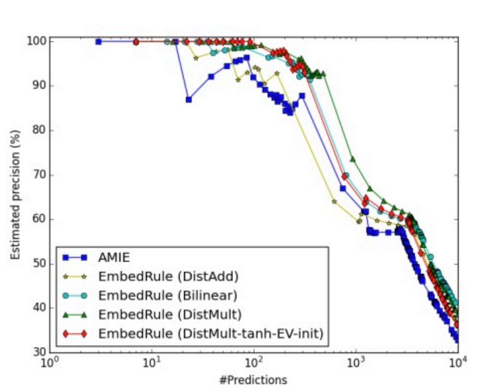


Figure 1: Aggregated precision of top length-2 rules extracted by different methods

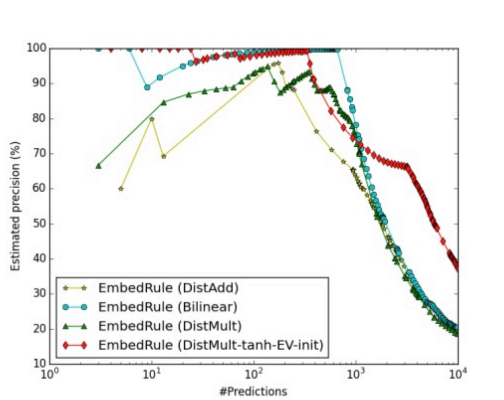


Figure 2: Aggregated precision of top length-3 rules extracted by different methods

- EMBEDRULE that uses embeddings trained from the bilinear objective
    - Suggests that the bilinear embeddings contain good amount of information about relations which allows for effective rule selection without looking at the entities
- Using multiplicative composition of matrix embeddings (from DISTMULT and BILINEAR) results in better performance compared to using additive composition of vector embeddings (from DISTADD)