Knowledge Graph Completion

高桢

2020年10月15日

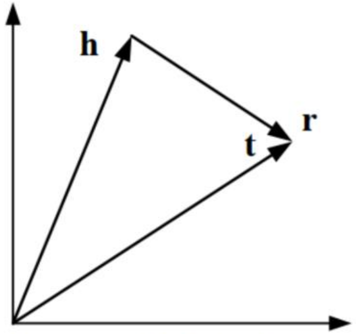
CONTENTS

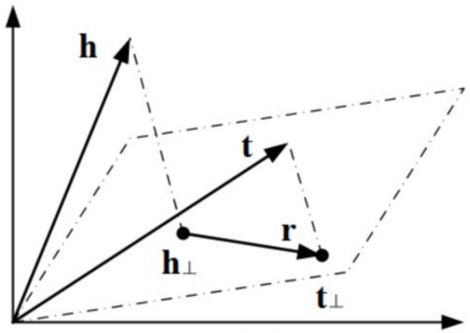
- Introduction
- KGE-based
  - A2N
  - TuckER
  - CapsE
- Rule-based
  - AnyBURL
- Application

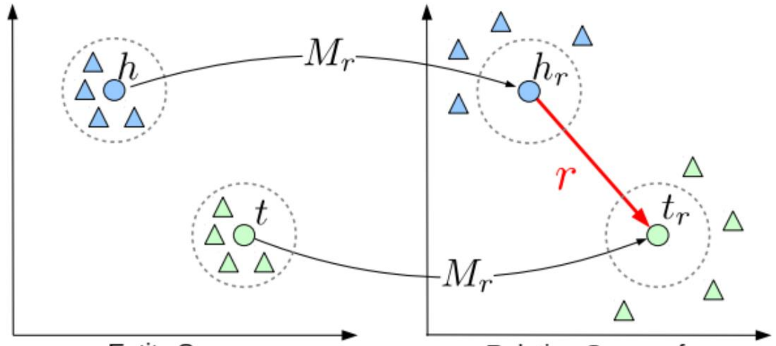
INTRODUCTION

- Knowledge
  - Triple (h, r, t)
  - h: head entity   r: relation   t: tail entity
  - Score (h, r, t)
    - Predict head entity
    - Predict relation
    - Predict tail entity
- Classify
  - Static KGC
  - Dynamic KGC

Translation Model







$$h + r \approx t$$

$$f_r(h,t) = ||h + r - t||_2^2$$

TransE

$$f_r(h,t) = ||h_\perp + r - t_\perp||_2^2$$

$$h_\perp = h - w_r^T h w_r$$

$$t_\perp = t - w_r^T t w_r$$

TransH

$$f_r(h,t) = ||h_\perp + r - t_\perp||_2^2$$

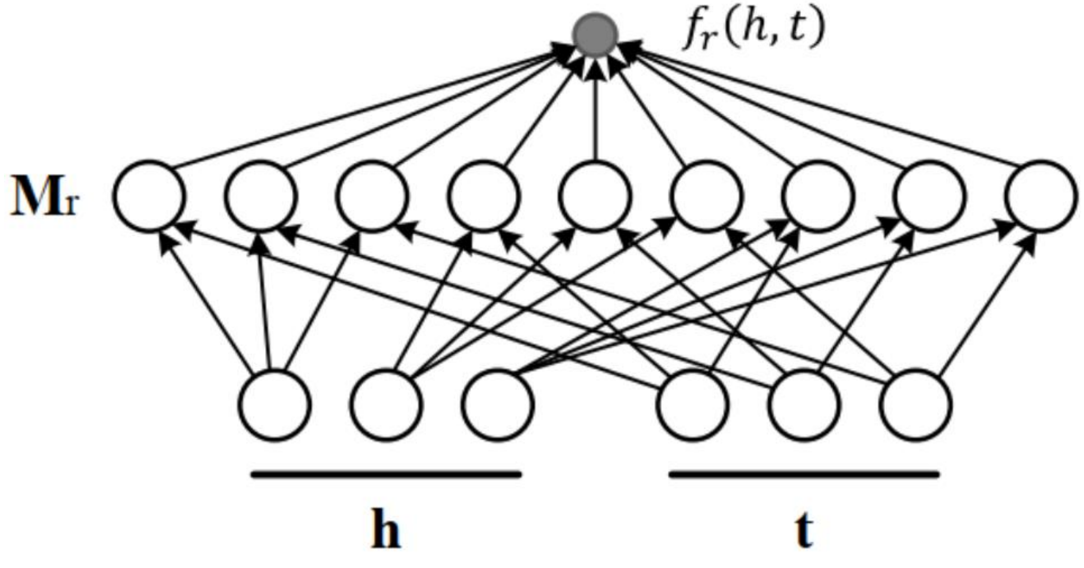
$$h_\perp = hM_r \qquad t_\perp = tM_r$$

TransR

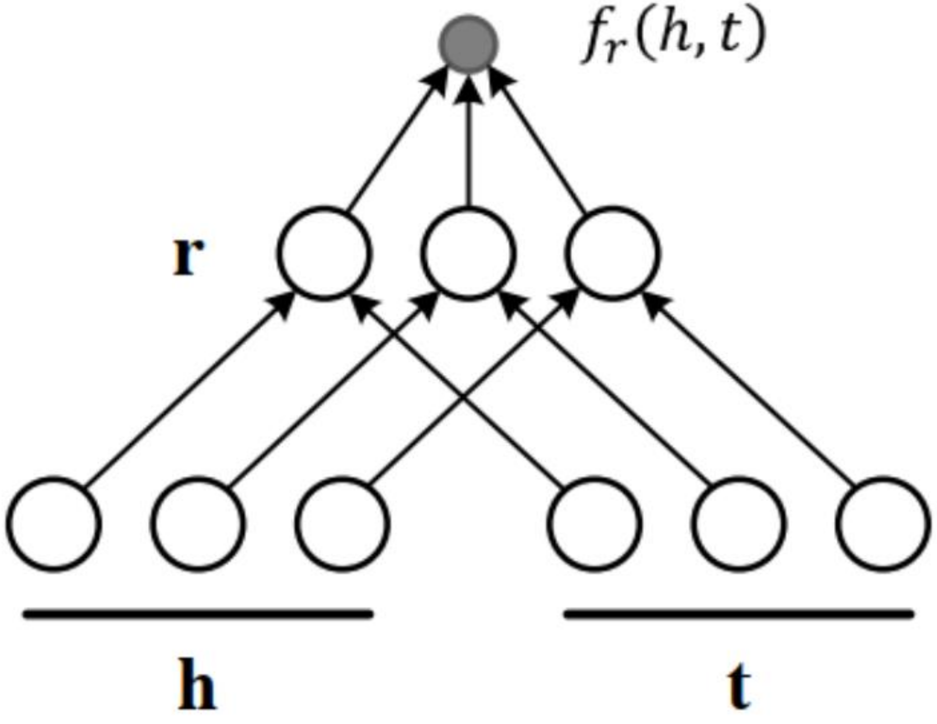
More Translation Model

Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h,t)$	Constraints/Regularization
TransE [14]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\\|\mathbf{h}+\mathbf{r}-\mathbf{t}\\|_{1/2}$	$\\|\mathbf{h}\\|_2=1, \\|\mathbf{t}\\|_2=1$
TransH [15]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^d$	$-\\|(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\\|_2^2$	$\\|\mathbf{h}\\|_2\le 1, \\|\mathbf{t}\\|_2\le 1$ $\|\mathbf{w}_r^\top\mathbf{r}\|/\\|\mathbf{r}\\|_2\le\epsilon, \\|\mathbf{w}_r\\|_2=1$
TransR [16]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r\in\mathbb{R}^{k\times d}$	$-\\|\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\\|_2^2$	$\\|\mathbf{h}\\|_2\le 1, \\|\mathbf{t}\\|_2\le 1, \\|\mathbf{r}\\|_2\le 1$ $\\|\mathbf{M}_r\mathbf{h}\\|_2\le 1, \\|\mathbf{M}_r\mathbf{t}\\|_2\le 1$
TransD [50]	$\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$ $\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^k$	$-\\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\\|_2^2$	$\\|\mathbf{h}\\|_2\le 1, \\|\mathbf{t}\\|_2\le 1, \\|\mathbf{r}\\|_2\le 1$ $\\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\\|_2\le 1$ $\\|(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\\|_2\le 1$
TranSparse [51]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r(\theta_r)\in\mathbb{R}^{k\times d}$ $\mathbf{M}_r^1(\theta_r^1), \mathbf{M}_r^2(\theta_r^2)\in\mathbb{R}^{k\times d}$	$-\\|\mathbf{M}_r(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M}_r(\theta_r)\mathbf{t}\\|_{1/2}^2$ $-\\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\\|_{1/2}^2$	$\\|\mathbf{h}\\|_2\le 1, \\|\mathbf{t}\\|_2\le 1, \\|\mathbf{r}\\|_2\le 1$ $\\|\mathbf{M}_r(\theta_r)\mathbf{h}\\|_2\le 1, \\|\mathbf{M}_r(\theta_r)\mathbf{t}\\|_2\le 1$ $\\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}\\|_2\le 1, \\|\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\\|_2\le 1$
TransM [52]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\theta_r\\|\mathbf{h}+\mathbf{r}-\mathbf{t}\\|_{1/2}$	$\\|\mathbf{h}\\|_2=1, \\|\mathbf{t}\\|_2=1$
ManifoldE [53]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-(\\|\mathbf{h}+\mathbf{r}-\mathbf{t}\\|_2^2-\theta_r^2)^2$	$\\|\mathbf{h}\\|_2\le 1, \\|\mathbf{t}\\|_2\le 1, \\|\mathbf{r}\\|_2\le 1$
TransF [54]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$(\mathbf{h}+\mathbf{r})^\top\mathbf{t}+(\mathbf{t}-\mathbf{r})^\top\mathbf{h}$	$\\|\mathbf{h}\\|_2\le 1, \\|\mathbf{t}\\|_2\le 1, \\|\mathbf{r}\\|_2\le 1$
TransA [55]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d, \mathbf{M}_r\in\mathbb{R}^{d\times d}$	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)^\top\mathbf{M}_r(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)$	$\\|\mathbf{h}\\|_2\le 1, \\|\mathbf{t}\\|_2\le 1, \\|\mathbf{r}\\|_2\le 1$ $\\|\mathbf{M}_r\\|_F\le 1, [\mathbf{M}_r]_{ij}=[\mathbf{M}_r]_{ji}\ge 0$
KG2E [45]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h)$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$ $\boldsymbol{\Sigma}_h,\boldsymbol{\Sigma}_t\in\mathbb{R}^{d\times d}$	$\mathbf{r}\sim\mathcal{N}(\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r)$ $\boldsymbol{\mu}_r\in\mathbb{R}^d, \boldsymbol{\Sigma}_r\in\mathbb{R}^{d\times d}$	$-\mathrm{tr}(\boldsymbol{\Sigma}_r^{-1}(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t))-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\mu}-\ln\frac{\det(\boldsymbol{\Sigma}_r)}{\det(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t)}$ $-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\ln(\det(\boldsymbol{\Sigma}))$ $\boldsymbol{\mu}=\boldsymbol{\mu}_h+\boldsymbol{\mu}_r-\boldsymbol{\mu}_t$ $\boldsymbol{\Sigma}=\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_r+\boldsymbol{\Sigma}_t$	$\\|\boldsymbol{\mu}_h\\|_2\le 1, \\|\boldsymbol{\mu}_t\\|_2\le 1, \\|\boldsymbol{\mu}_r\\|_2\le 1$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_h\le c_{max}\mathbf{I}$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_t\le c_{max}\mathbf{I}$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_r\le c_{max}\mathbf{I}$

RESCAL & DisMult



$$f_r(h,t) = h^T M_r t$$



$$f_r(h,t) = h^T diag(M_r) t$$

A2N: Attending to Neighbors for Knowledge Graph Inference

ACL 2019

Abstract

- Attention-based model
- Query-dependent representation
- Calculate by graph neighborhood of entity
- Interpretable

[1]A2N: Attending to Neighbors for Knowledge Graph Inference. ACL2019

Formula

$$\tilde{n}_i = W_n[\tilde{r}_i; \tilde{e}_i^0] \quad W_n \in \mathbb{R}^{K \times 2K}$$

$$a_i = f(s, r, n_i) = (\tilde{s}^0)^T Diag(\tilde{r}) \tilde{n}_i$$

$$p_i = \frac{\exp(a_i)}{\sum_{j \leq |N_s|} \exp(a_j)}$$

$$\hat{s} = \sum_{i \leq |N_s|} p_i \tilde{n}_i$$

$$\tilde{s} = W_s[\hat{s}; \tilde{s}^0] \quad W_s \in \mathbb{R}^{K \times 2K}$$

DistMult score:

$$f(s, r, t) = \tilde{s}^T Diag(\tilde{r}) \tilde{t}$$

[1]A2N: Attending to Neighbors for Knowledge Graph Inference. ACL2019

Interpretable

(Bill_Payne, profession, ?)
Prediction: Musician
Top Neighbors:
(recording_contribution, Synthesizer) Prob: 0.0911
(Inverse: Instrumentalist, Keyboards) Prob: 0.0906
(track_contribution, Synthesizer) Prob: 0.0878
(Inverse: Instrumentalist, Hammond_organ) Prob: 0.0823
(track_contribution, Accordion) Prob: 0.0758

(Burt_Young, nationality, ?)
Prediction: US
Top Neighbors:
(place_of_birth, Queens) Prob: 0.2714
(places_lived, Queens) Prob: 0.2039
(Inverse: ethnicity/people, Italian_American) Prob: 0.1860
(performance/film, Transamerica) Prob: 0.0445
(gender, Male) Prob: 0.0372

(Fantastic_Four:_Rise_of_the_Silver_Surfer, genre, ?)
Prediction: Fantasy
Top Neighbors:
(genre, Superhero_film) Prob: 0.0614
(genre, Superhero) Prob: 0.0490
(genre, Science_fiction_film) Prob: 0.0460
(genre, Action_film) Prob: 0.0443
(language, Arabic_language) Prob: 0.0395

(Armstrong_County,_Pennsylvania, time_zones, ?)
Prediction: Eastern_Time_Zone
Top Neighbors:
(Inverse: location/contains, Pittsburgh_metropolitan_area) Prob: 0.3219
(Inverse: location/contains, Pennsylvania) Prob: 0.2994
(Inverse: location/country/second_level_divisions, US) Prob: 0.1092
(estimated_number_of_mortages/source, US_Department_of_HUD) Prob: 0.0692
(currency, US_dollar) Prob: 0.0309

[1]A2N: Attending to Neighbors for Knowledge Graph Inference. ACL2019

Experiments

	FB15k-237				WN18RR			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
DistMult	0.370	0.568	0.417	0.275	0.43	0.48	0.44	0.41
ComplEx	0.394	0.572	0.434	0.303	0.42	0.48	0.43	0.38
ConvE	0.410	0.600	0.457	0.313	0.44	0.52	0.45	0.40
MINERVA	0.293	0.456	0.329	0.217	0.45	0.51	0.46	0.41
A2N	**0.422**	**0.608**	**0.464**	**0.328**	**0.49**	**0.55**	**0.50**	**0.45**

Table 1: Results for target-only prediction of various models. A2N performs significantly better.

	FB15k-237				WN18RR			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
DistMult	0.278	0.444	0.304	0.196	0.43	0.49	0.44	0.39
ComplEx	0.247	0.428	0.275	0.158	0.44	0.51	0.46	0.41
R-GCN	0.249	0.417	0.264	0.151	–	–	–	–
ConvE	**0.325**	**0.501**	**0.356**	**0.237**	0.43	**0.52**	0.44	0.40
A2N	0.317	0.486	0.348	0.232	**0.45**	0.51	**0.46**	**0.42**

Table 2: Results for both source and target prediction of various models. A2N performs better or competitively to most state-of-the-art models, specially on top prediction (Hits@1).

[1]A2N: Attending to Neighbors for Knowledge Graph Inference. ACL2019

A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization

NAACL2019

Capsule Network

- Capsule like neuron
- Capsule output a vector rather than value
- Neuron only detects a special pattern
- Judge by the length of the output vector

[2] A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. NAACL2019

CapsE Model

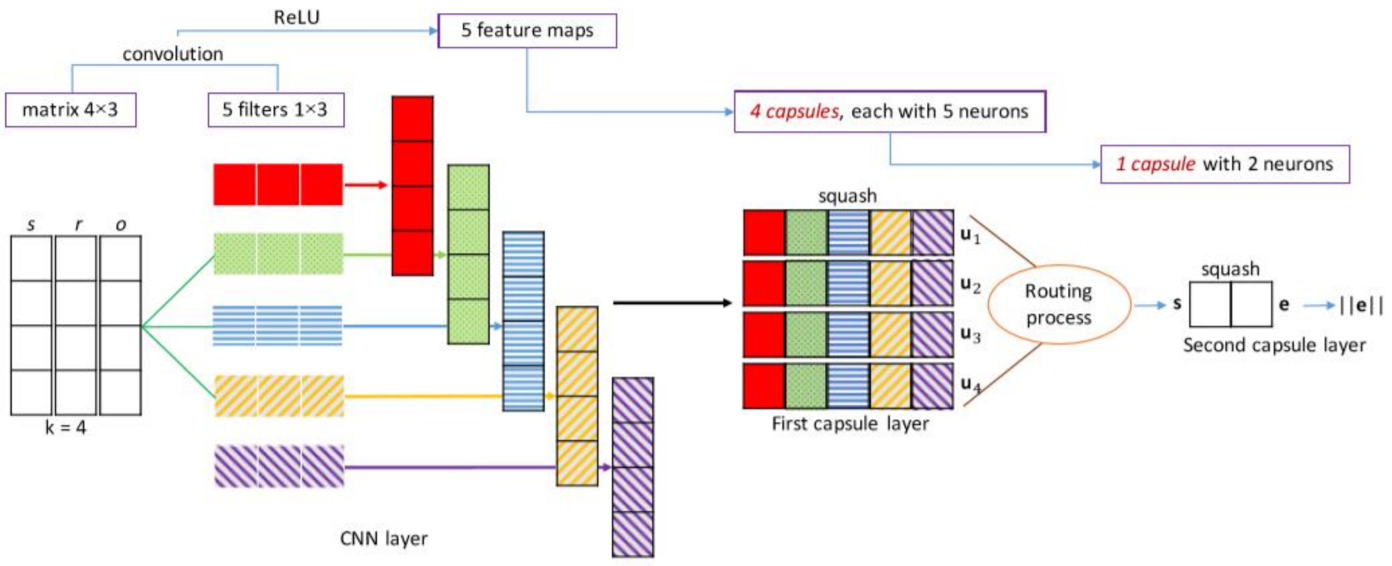


Figure 1: An example illustration of our CapsE with $k = 4$, N $= 5$, and $d = 2$.

for *all capsule* $i \in$ *the first layer* **do**
  $b_i \leftarrow 0$

for iteration = $1, 2, \ldots, m$ **do**
  $\mathbf{c} \leftarrow \mathsf{softmax}\,(\mathbf{b})$
  $\mathbf{s} \leftarrow \sum_i c_i \hat{\mathbf{u}}_i$
  $\mathbf{e} = \mathsf{squash}\,(\mathbf{s})$
  for *all capsule* $i \in$ *the first layer* **do**
    $b_i \leftarrow b_i + \hat{\mathbf{u}}_i \cdot \mathbf{e}$

$$\mathsf{squash}\,(\mathbf{s}) = \frac{\|\mathbf{s}\|^2}{1+\|\mathbf{s}\|^2}\frac{\mathbf{s}}{\|\mathbf{s}\|}$$

[2] A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. NAACL2019

TuckER: Tensor Factorization for Knowledge Graph Completion

ICML2019

Tucker decomposition

$$\mathcal{X} \in \mathbb{R}^{I \times J \times K}$$

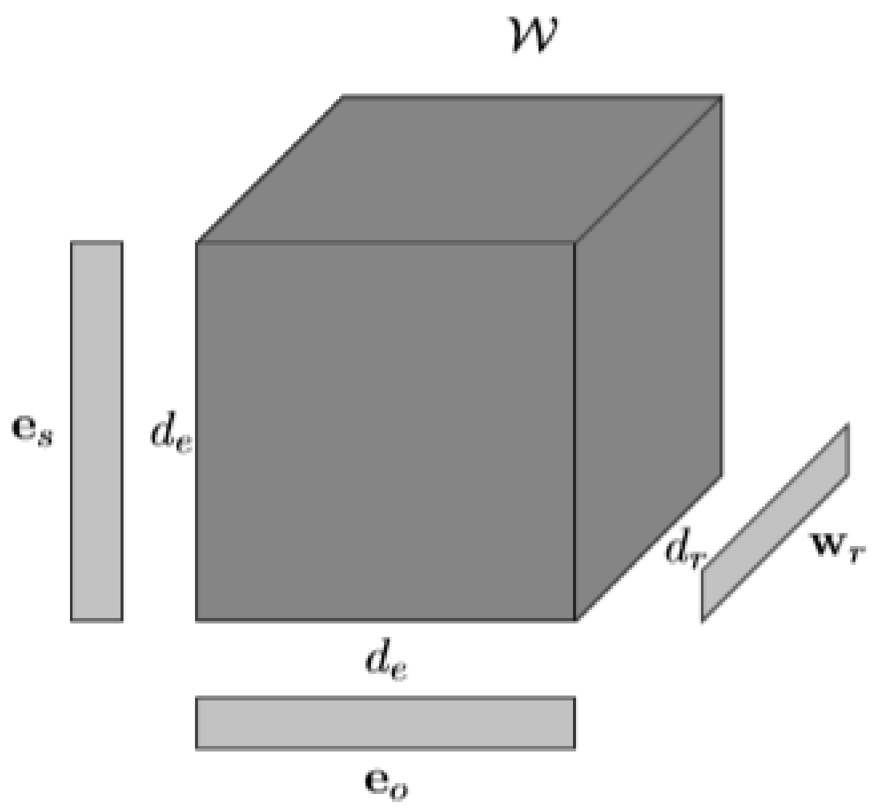
$$\mathcal{X} \approx \mathcal{Z} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

$$\mathcal{Z} \in \mathbb{R}^{P \times Q \times R}$$ Core tensor

$$\mathbf{A} \in \mathbb{R}^{I \times P} \quad \mathbf{B} \in \mathbb{R}^{J \times Q} \quad \mathbf{C} \in \mathbb{R}^{K \times R}$$

[3] TuckER: Tensor Factorization for Knowledge Graph Completion

TuckER Architecture



$$\mathbf{E} = \mathbf{A} = \mathbf{C} \in \mathbb{R}^{n_e \times d_e}$$

$$\mathbf{R} = \mathbf{B} \in \mathbb{R}^{n_r \times d_r}$$

$\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^{d_e}$ Rows of entity embedding matrix **E**

$\mathbf{w}_r \in \mathbb{R}^{d_r}$ Rows of relation embedding matrix **R**

$\mathcal{W} \in \mathbb{R}^{d_e \times d_r \times d_e}$ Core tensor

$$\phi(e_s, r, e_o) = \mathcal{W} \times_1 \mathbf{e}_s \times_2 \mathbf{w}_r \times_3 \mathbf{e}_o$$

[3] TuckER: Tensor Factorization for Knowledge Graph Completion

Advantages

- Fully expressive
- Entity and relation embedding dimension determines number of parameters
- RESCAL, DistMult can be viewed as a special case of TuckER
  - RESCAL:

$$I = K = n_e \quad P = R = d_e \quad Q = J = n_r \quad \mathbf{B} = \mathbf{I}_J$$

  - DistMult:

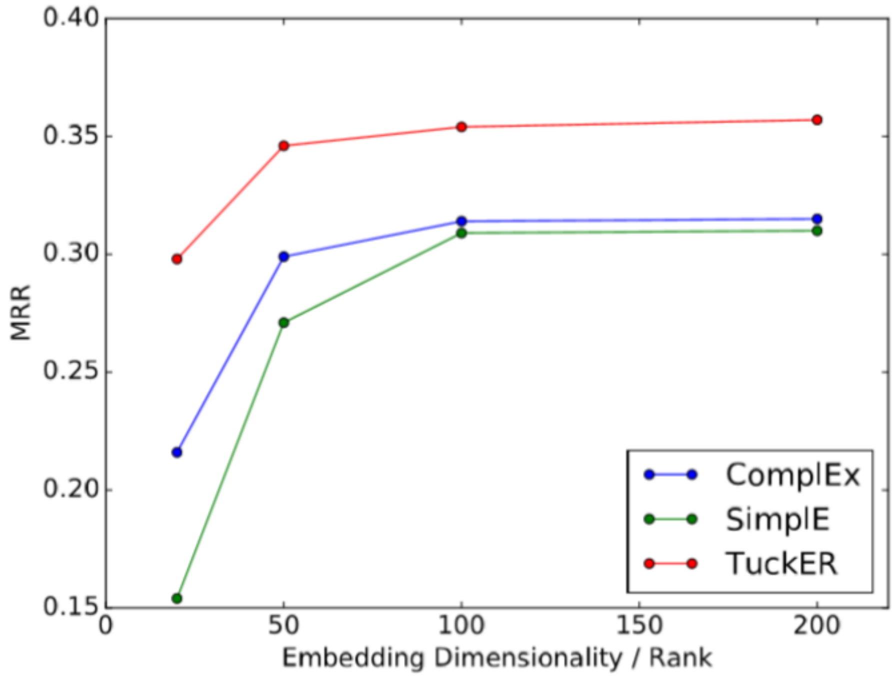
$$P = Q = R = d_e \qquad z_{pqr} = \begin{cases} 1, p = q = r \\ 0, p \neq \mathrm{q} \neq \mathrm{r} \end{cases}$$

[3] TuckER: Tensor Factorization for Knowledge Graph Completion

Experiments

	Linear	WN18 MRR	WN18 Hits@10	WN18 Hits@3	WN18 Hits@1	FB15k MRR	FB15k Hits@10	FB15k Hits@3	FB15k Hits@1
TransE (Bordes et al., 2013)	no	–	.892	–	–	–	.471	–	–
DistMult (Yang et al., 2015)	yes	.822	.936	.914	.728	.654	.824	.733	.546
ComplEx (Trouillon et al., 2016)	yes	.941	.947	.936	.936	.692	.840	.759	.599
ANALOGY (Liu et al., 2017)	yes	.942	.947	.944	.939	.725	.854	.785	.646
Neural LP (Yang et al., 2017)	no	.940	.945	–	–	.760	.837	–	–
R-GCN (Schlichtkrull et al., 2018)	no	.819	**.964**	.929	.697	.696	.842	.760	.601
TorusE (Ebisu and Ichise, 2018)	no	.947	.954	.950	.943	.733	.832	.771	.674
ConvE (Dettmers et al., 2018)	no	.943	.956	.946	.935	.657	.831	.723	.558
HypER (Balažević et al., 2019)	no	.951	958	**.955**	.947	.790	.885	.829	.734
SimplE (Kazemi and Poole, 2018)	yes	.942	.947	.944	.939	.727	.838	.773	.660
TuckER (ours)	yes	**.953**	.958	**.955**	**.949**	**.795**	**.892**	**.833**	**.741**

Table 4: Link prediction results on WN18 and FB15k.



[3] TuckER: Tensor Factorization for Knowledge Graph Completion

Anytime Bottom-Up Rule Learning for Knowledge Graph Completion

IJCAI19

AnyBURL

Algorithm 1 Anytime Bottom-up Rule Learning

```
AnyBURL(𝔾, s, sat, Q, ts)
n = 2
R = ∅
loop
    R_s = ∅
    start = currentTime()
    repeat
        p = samplePath(𝔾, n)
        R_p = generateRules(p)
        for r ∈ R_p do
            score(r, s)
            if Q(r) then
                R_s = R_s ∪ {r}
            end if
        end for
    until currentTime() > start + ts
    R'_s = R_s ∩ R
    if |R'_s|/|R_s| > sat then
        n = n + 1
    end if
    R = R_s ∪ R
end loop
return R
```

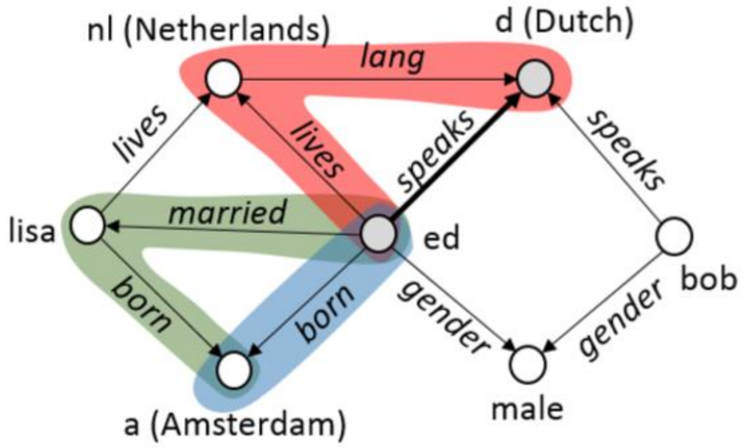


Figure 1: A small knowledge graph $\mathbb{G}$ used for sampling paths. We marked the body of Rule 1 (blue), Rule 2 (green), and Rule 3 (red).

$$speaks(ed, d) \leftarrow born(ed, a) \quad (1)$$

$$speaks(ed, d) \leftarrow married(ed, lisa), born(lisa, a) \quad (2)$$

$$speaks(ed, d) \leftarrow lives(ed, nl), lang(nl, d) \quad (3)$$

[4] Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. IJCAI19

Advantages

- Candidate ranking can be explained
- Some rules can be reused
- Not require to learn dataset specific hyper parameters

[4] Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. IJCAI19

Application

- Fact judgement
- Entity classification
- Automatically check the quality of relation extraction