Графы знаний

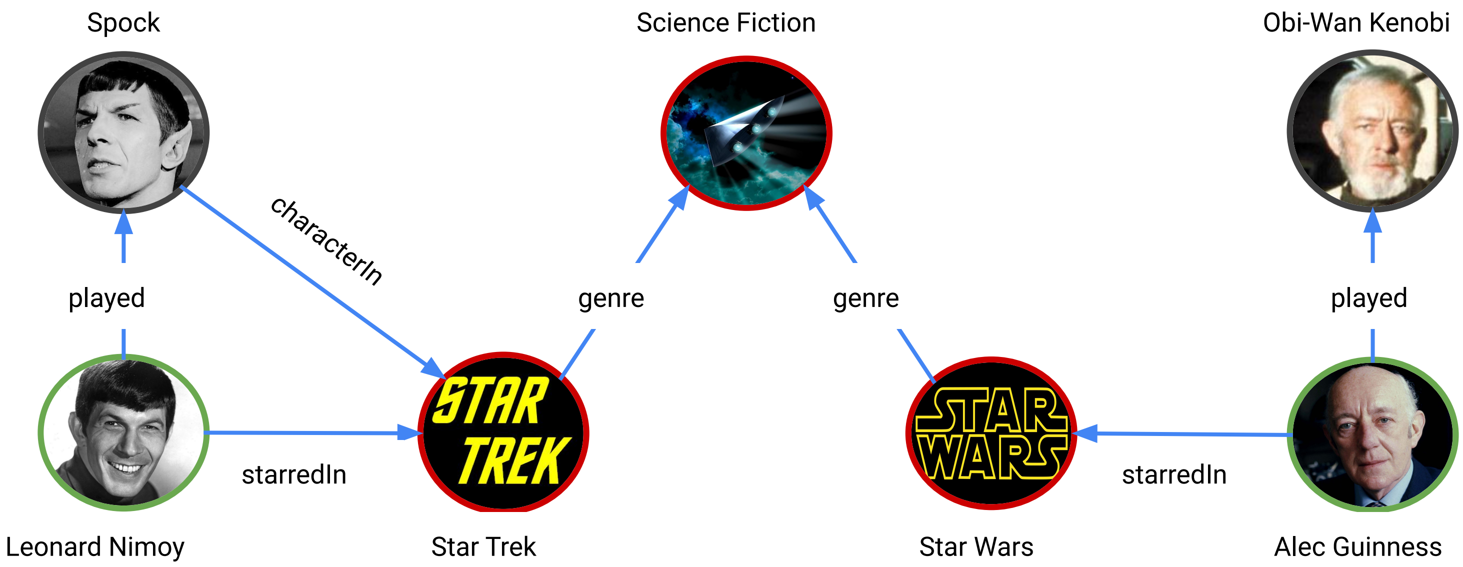
Лекция 7 - Knowledge Graph Embeddings

М. Галкин, Д. Муромцев

Сегодня

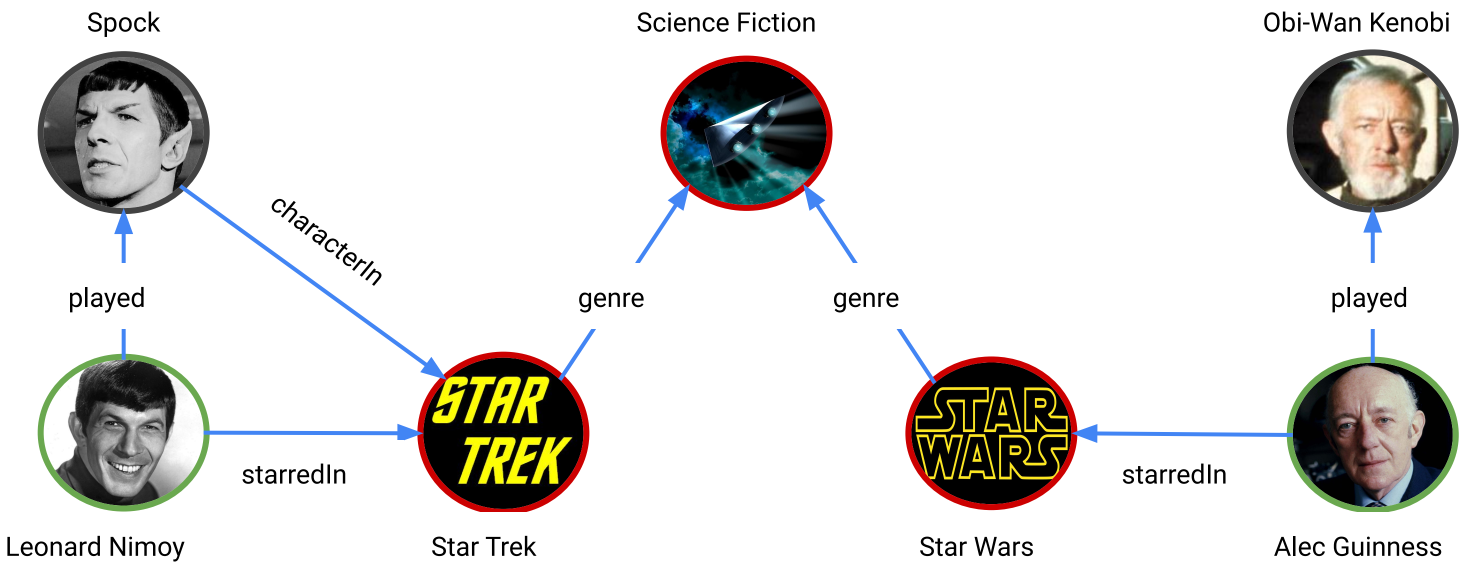
1. Introduction
2. Представление знаний в графах - RDF & RDFS & OWL
3. Хранение знаний в графах - SPARQL & Graph Databases
4. Однородность знаний - RDF* & Wikidata & SHACL & ShEx
5. Интеграция данных в графы знаний - Semantic Data Integration
6. Введение в теорию графов - Graph Theory Intro
7. **Векторные представления графов - Knowledge Graph Embeddings**
8. Машинное обучение на графах - Graph Neural Networks & KGs
9. Некоторые применения - Question Answering & Query Embedding

Представление знаний - онтологическое




```
LeonardNimoy   starredIn     StarTrek;        AlecGuinness   starredIn     StarWars;
               played        Spock.                          played        Obi-Wan.
Spock          characterIn   StarTrek .       StarWars       genre         SciFi .
```

Представление знаний - статистическое

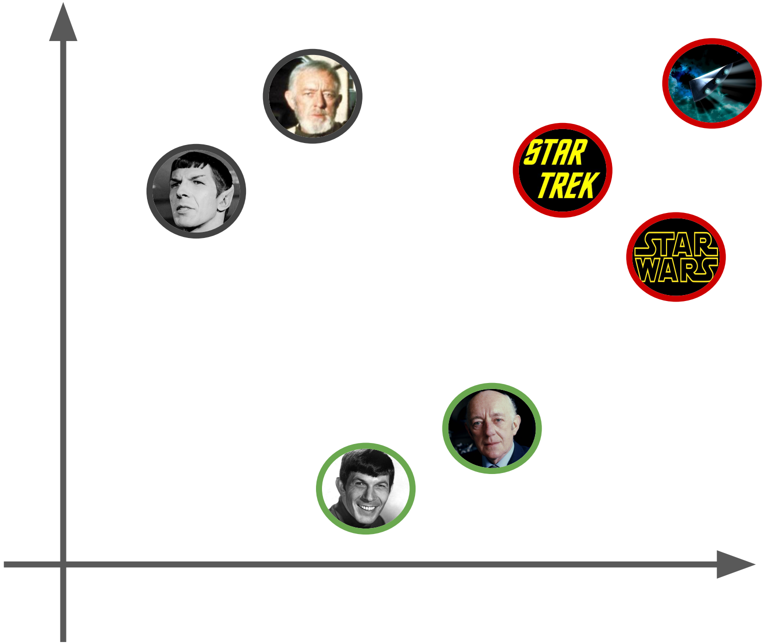


Spock = [0.1, 0.2, 0.3]
Leonard Nimoy = [0.4, 0.8, 0.1]
Star Trek = [0.22, 0.34, 0.87]

characterIn = [0.1, 0.1, 0.6]

Obi-Wan = [0.05, 0.25, 0.37]
Alec Guinness = [0.33, 0.5, 0.3]
Star Wars = [0.18, 0.4, 0.9]

Embedding - что мы хотим получить

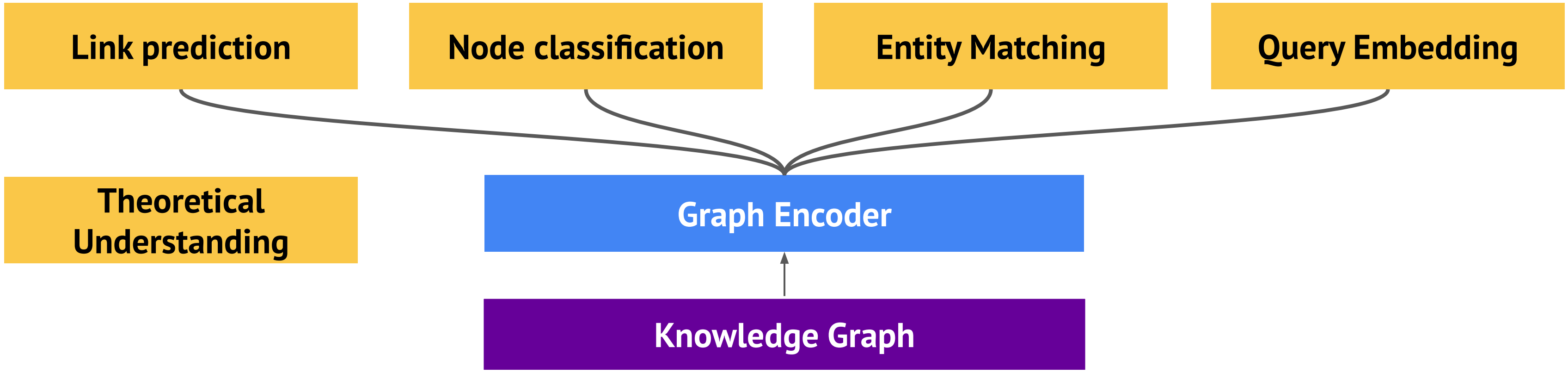


Big Picture in $\mathbb{R}^5$

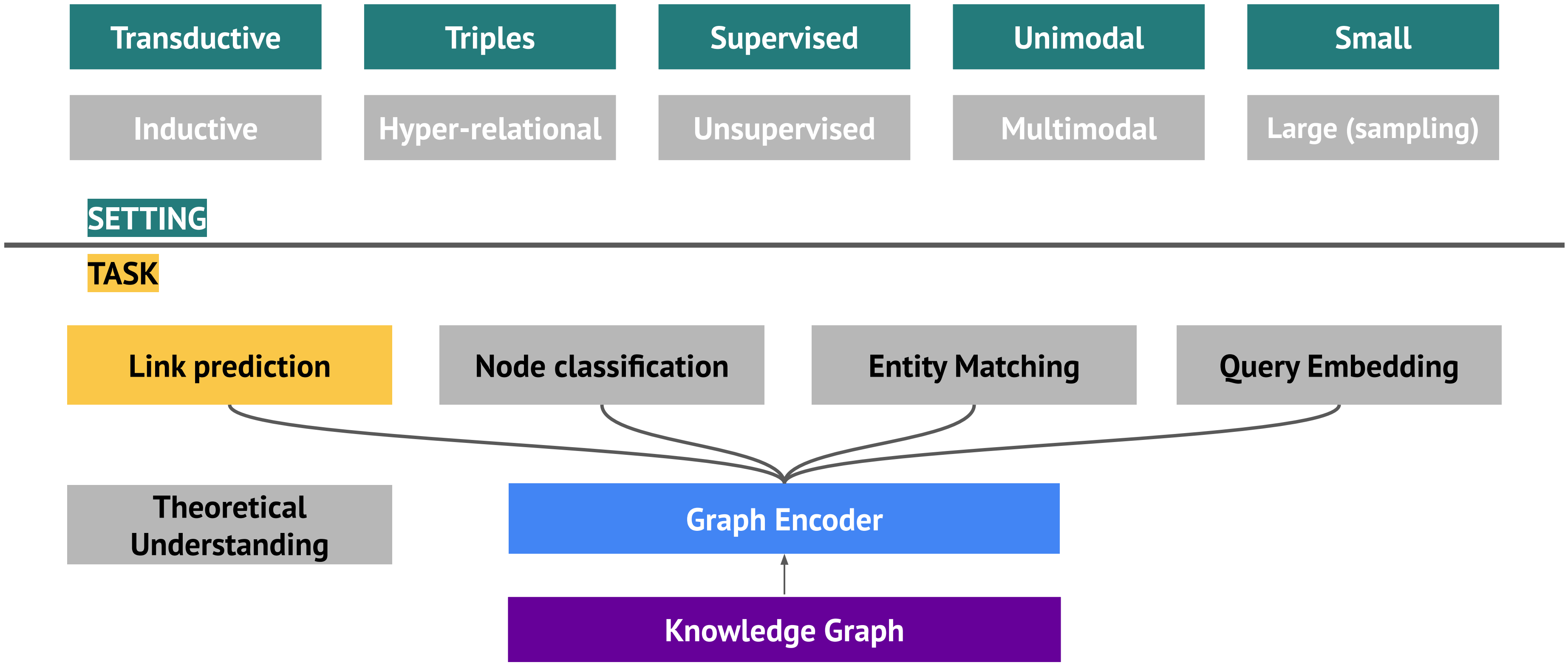
Transductive	Triples	Supervised	Unimodal	Small
Inductive	Hyper-relational	Unsupervised	Multimodal	Large (sampling)

SETTING

TASK



В этой лекции



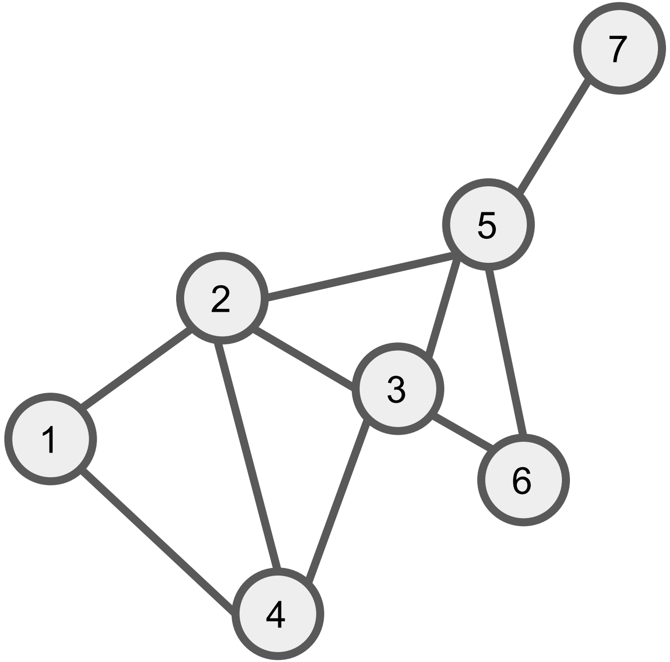
В этой лекции

Transductive	Весь граф известен во время тренировки
Triples	KG состоит только из триплетов (про RDF* - позднее) **без литералов**
Supervised	Обучение с учителем на конкретной задаче. Сигнал - известные связи
Unimodal	Только граф, без текста / видео / изображений / геометрии / другого
Small	Стандартные датасеты - до 100K узлов

Почему бы не взять методы из прошлой лекции?

Классический граф

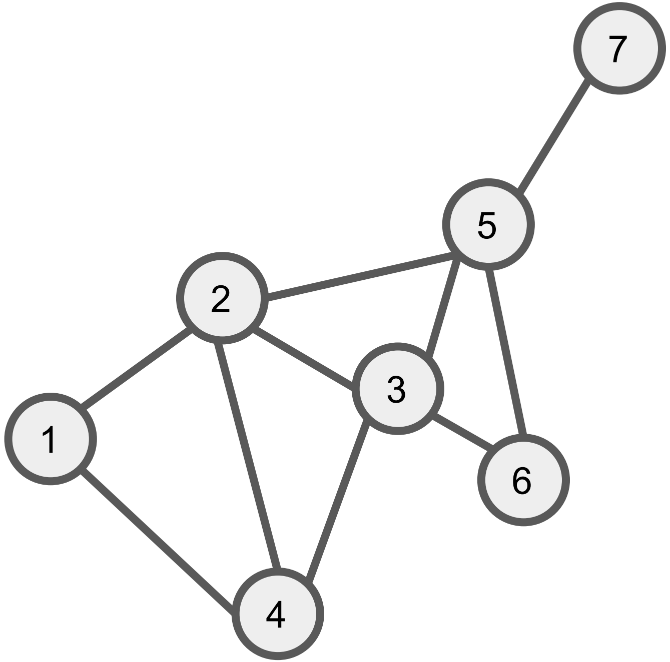
- Ненаправленный
- Нет типов связей



Почему бы не взять методы из прошлой лекции?

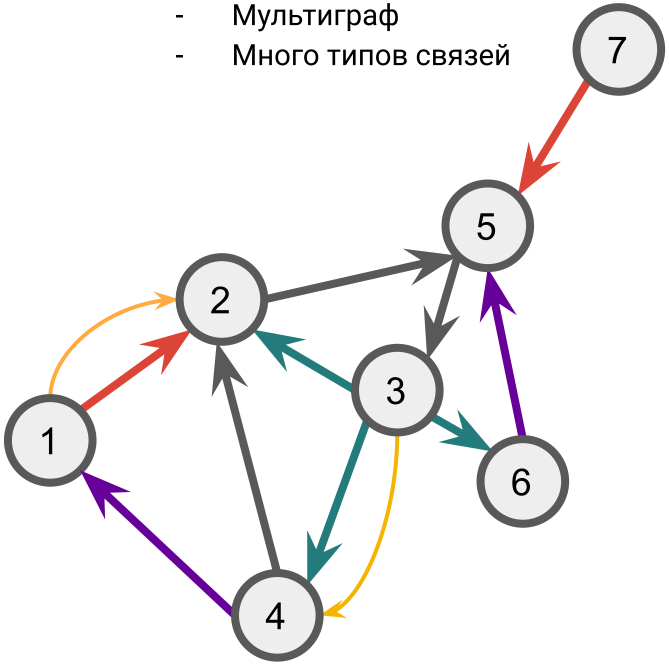
Классический граф

- Ненаправленный
- Нет типов связей

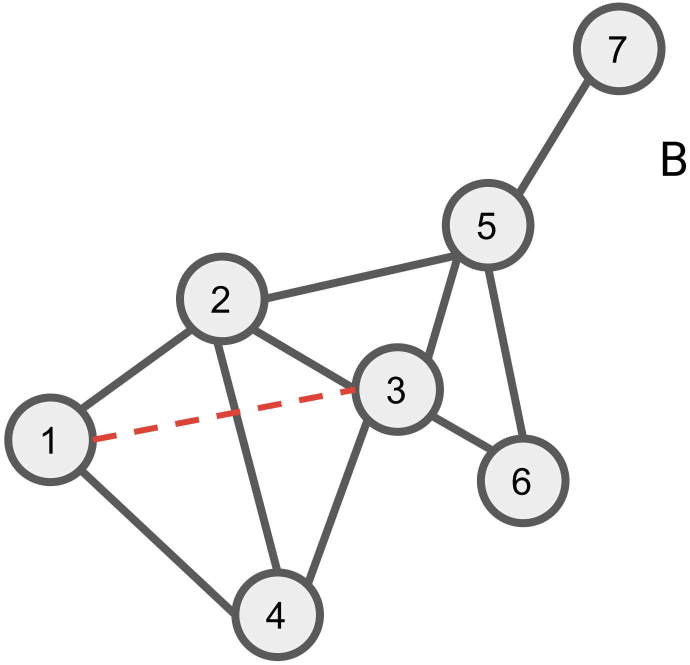


Наш клиент

- Направленный
- Мультиграф
- Много типов связей



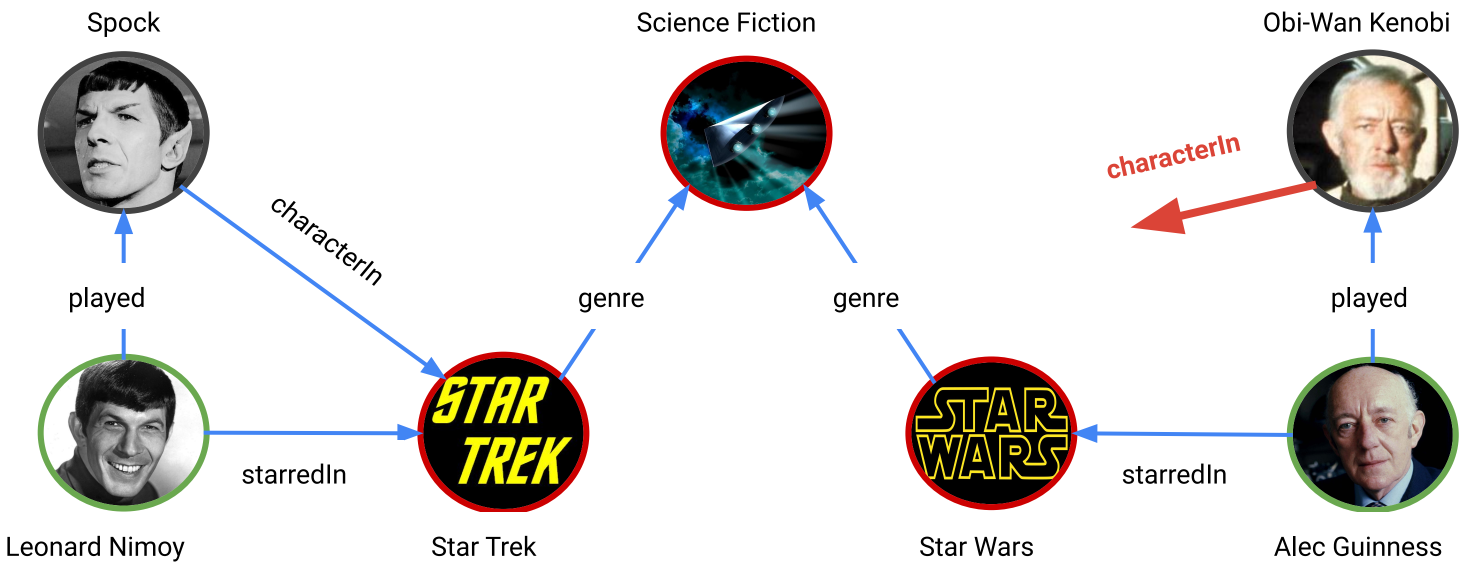
Классический Link Prediction



Вероятность связи как функция от вершин

$$p(1, 3) = f(1, 3)$$

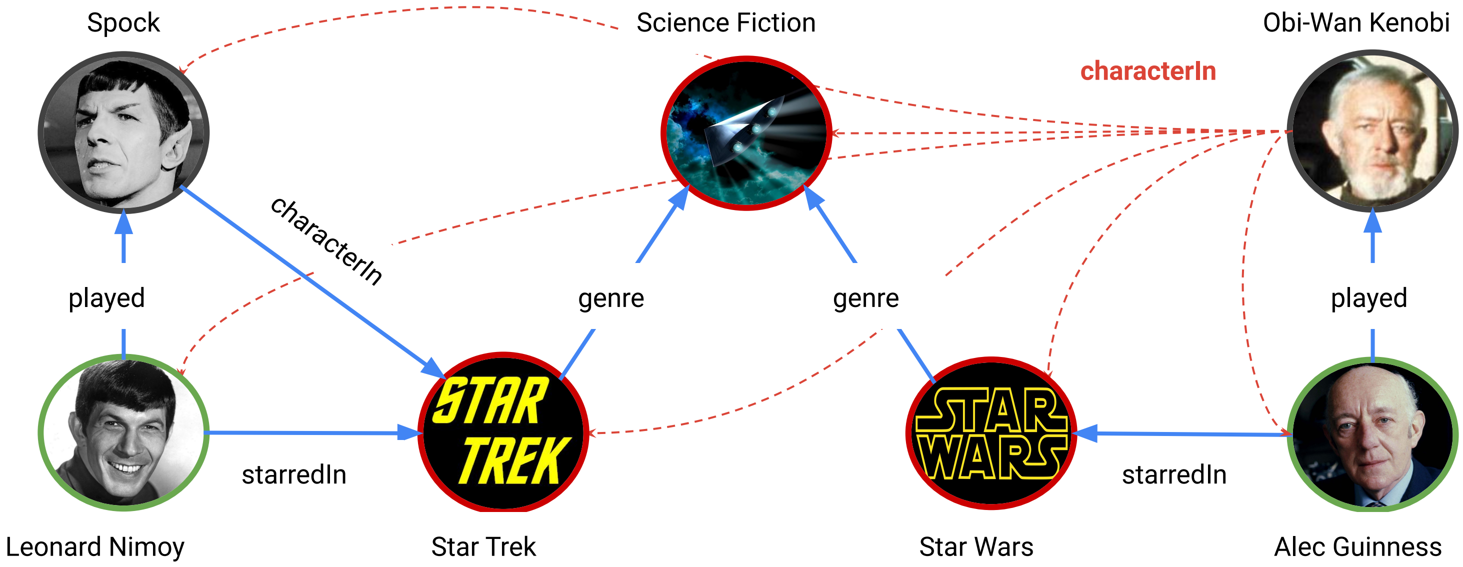
KG Link Prediction (KG Completion)



(Obi-Wan, characterIn, ?)

Nickel et al. A review of relational machine learning for knowledge graphs. IEEE. 2015

KG Link Prediction (KG Completion)

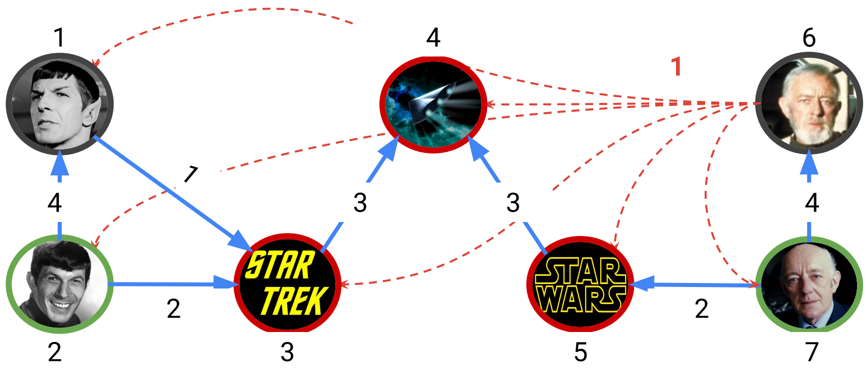


$$query = (head, relation, ?)$$

$$p_{hrt} \approx score(h, r, t)$$

Nickel et al. A review of relational machine learning for knowledge graphs. IEEE. 2015

KG Link Prediction - Input



e2id
1: Spock
2: Leonard Nimoy
3: Star Trek
4: SciFi
5: Star Wars
6: Obi-Wan
7: Alec Guinness

r2id
1: characterIn
2: starredIn
3: genre
4: played

1	1	3
2	4	1
2	2	3
3	3	4
5	3	4
7	2	5
7	4	6
6	1	?

Типичный вход KG embedding моделей

Relational Patterns

$r(h, t) = r(t, h)$ Симметричность

```
knows(a, b) -> knows(b, a)
```

$r(h, t) \neq r(t, h)$ Антисимметричность

```
friend(a, b) -> ¬friend(b, a)
```

$r_1(h, t) = r_2(t, h)$ Инверсия

```
cast(a, b) -> starredIn(b, a)
```

$r_1(h, t) \wedge r_2(t, z) \rightarrow r_3(h, z)$ Композиция

```
mother(a, b) ∧ husband(b, c) -> father(a, c)
```

$r(h, t_1), r(h, t_2), \ldots, r(h, t_n)$ Отношения 1-N

```
hasCity(country, city1),
hasCity(country, city2), ...
```

Knowledge Graph Embeddings

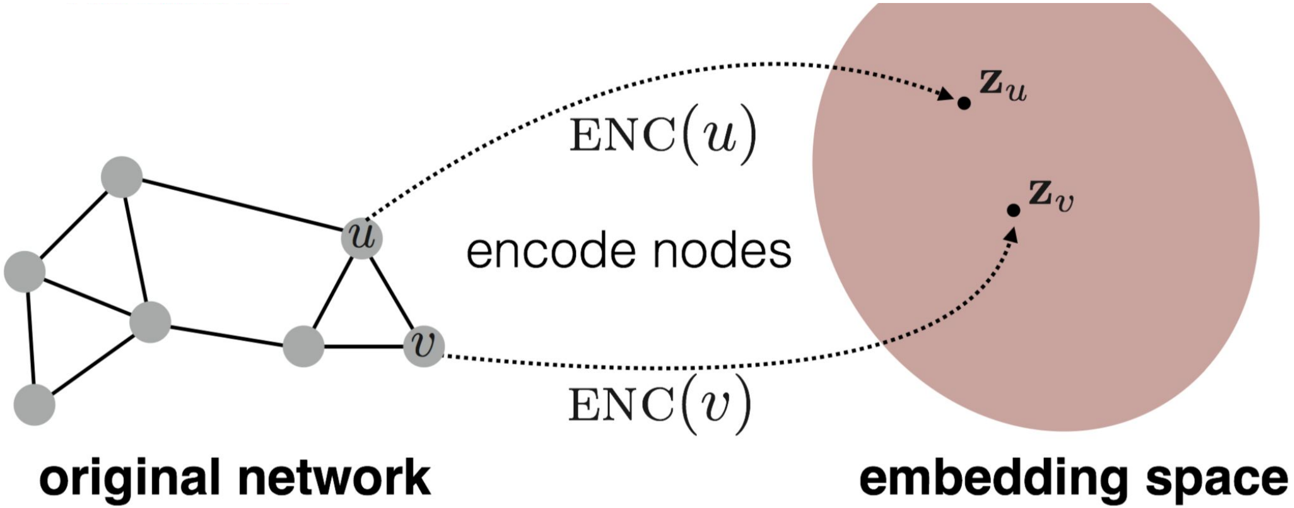
Tensor Factorization

Translation

Neural Networks

Graph Neural Nets

Goal: encode nodes so that **similarity in the embedding space (e.g., dot product)** approximates **similarity in the original network**



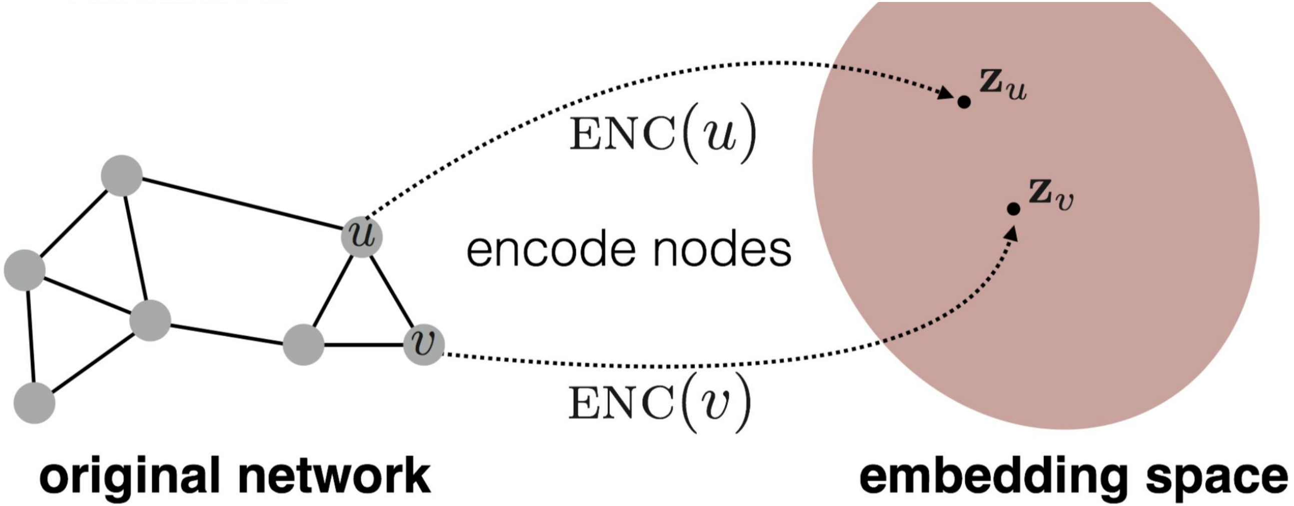
Source: Stanford CS224w, http://web.stanford.edu/class/cs224w/

Knowledge Graph Embeddings

Tensor Factorization

Translation

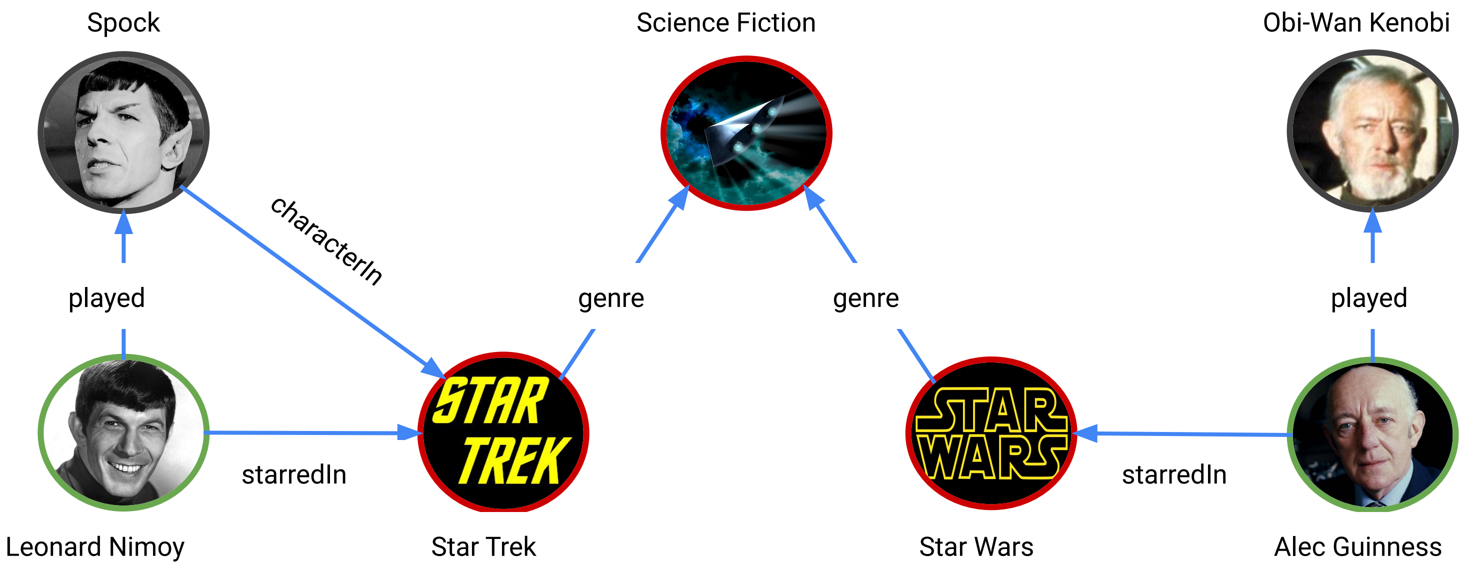
Neural Networks



Shallow embedding
(каждый узел и тип связи -> уникальный вектор)

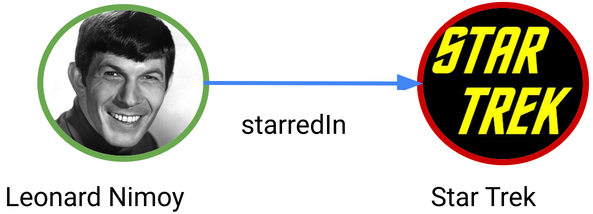
Source: Stanford CS224w, http://web.stanford.edu/class/cs224w/

KGE - Graphs as Tensors

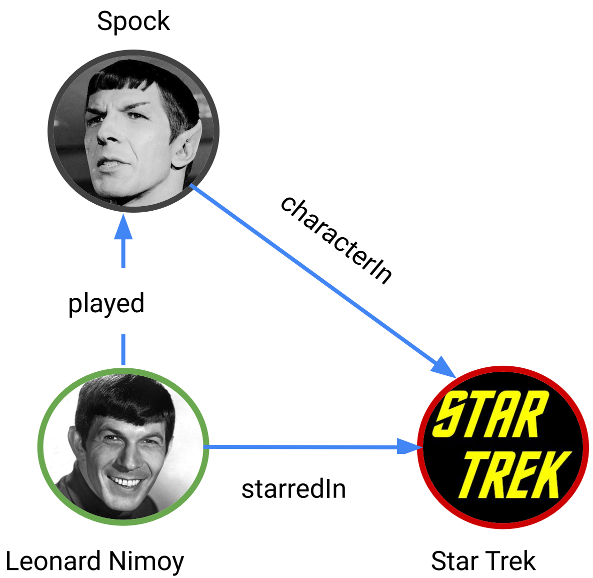


Nickel et al. A review of relational machine learning for knowledge graphs. IEEE. 2015

KGE - Graphs as Tensors



KGE - Graphs as Tensors



	0	1	0
	0	0	0
	0	0	0

starredIn

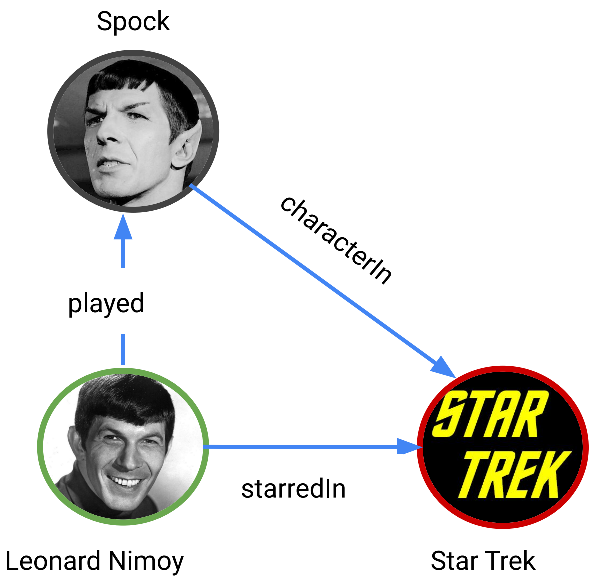
	0	0	1
	0	0	0
	0	0	0

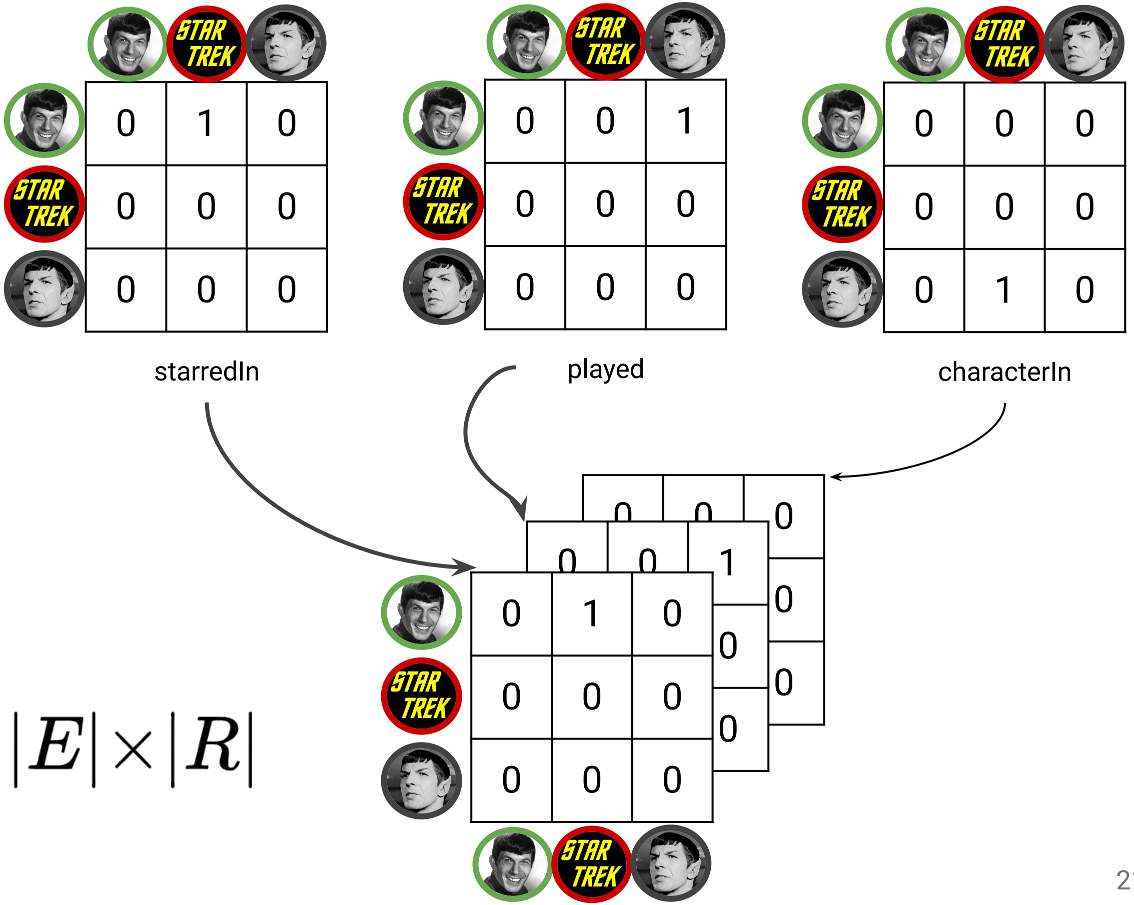
played

	0	0	0
	0	0	0
	0	1	0

characterIn

KGE - Graphs as Tensors



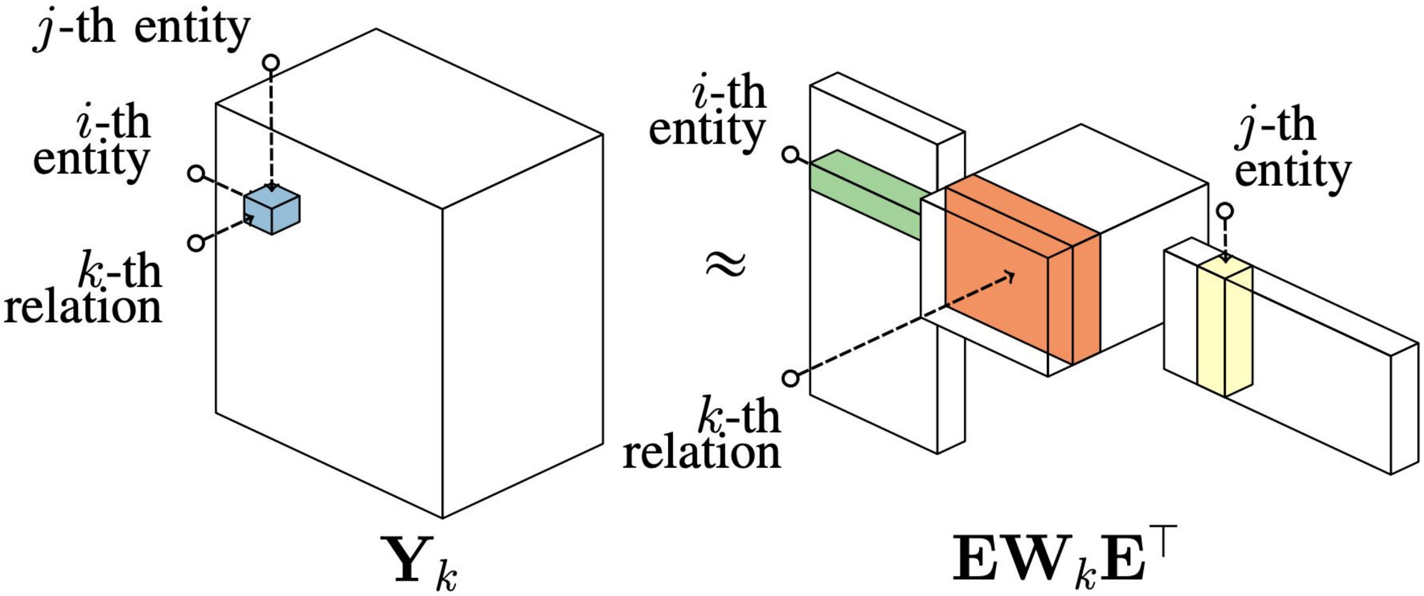


$$\mathcal{T} : \mathbb{R}^{|E| \times |E| \times |R|}$$

KGE - RESCAL

Tensor Factorization

Goal - factorize a sparse 3D tensor to dense E and R

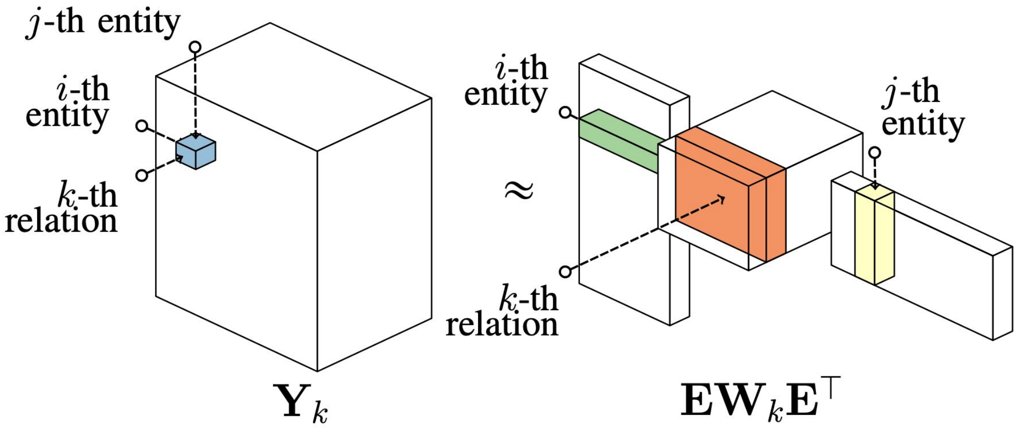


Nickel et al. A review of relational machine learning for knowledge graphs. IEEE. 2015

KGE - RESCAL

Tensor Factorization

Goal - factorize a sparse 3D tensor to dense E and R



$$\mathbf{E} : \mathbb{R}^{|E| \times n} \qquad \mathbf{W} : \mathbb{R}^{|k| \times n \times n}$$

Nickel et al. A review of relational machine learning for knowledge graphs. IEEE. 2015

KGE - RESCAL

Tensor Factorization

Goal - factorize a sparse 3D tensor to dense E and R

	Entity matrix	Relations matrix	Score function (e1,r,e2)
RESCAL	$\mathbf{E} : \mathbb{R}^{\|E\| \times n}$	$\mathbf{W} : \mathbb{R}^{\|k\| \times n \times n}$	$e_1 \cdot W_r \cdot e_2$

Yang et al. Embedding Entities And Relations For Learning And Inference In Knowledge Bases. ICLR 2015

KGE - DistMult

Tensor Factorization

Goal - factorize a sparse 3D tensor to dense E and R

	Entity matrix	Relations matrix	Score function (e1,r,e2)
RESCAL	$\mathbf{E} : \mathbb{R}^{\lvert E \rvert \times n}$	$\mathbf{W} : \mathbb{R}^{\lvert k \rvert \times n \times n}$	$e_1 \cdot W_r \cdot e_2$
		↓ diag(Wk) ↓	
DistMult	$\mathbf{E} : \mathbb{R}^{\lvert E \rvert \times n}$	$\mathbf{W} : \mathbb{R}^{\lvert k \rvert \times n}$	$e_1 \cdot r \cdot e_2$

Yang et al. Embedding Entities And Relations For Learning And Inference In Knowledge Bases. ICLR 2015

KGE - DistMult & Patterns

Tensor Factorization

$$h \cdot r \cdot t$$

Симметричность

$$h \cdot r \cdot t = t \cdot r \cdot h$$

Инверсия

$$h \cdot r_1 \cdot t = t \cdot r_2 \cdot h \rightarrow r_1 = r_2$$

Антисимметричность

Умножение коммутативно

Композиция

Невозможно геометрически

Отношения 1-N

$$h \cdot r \cdot t_1 \neq h \cdot r \cdot t_n$$

KGE - ComplEx

Tensor Factorization

ComplEx - let’s use complex numbers instead of real

	Entity matrix	Relations matrix	Score function (e1,r,e2)
DistMult	$\mathrm{E} : \mathbb{R}^{\lvert E \rvert \times n}$	$\mathbf{W} : \mathbb{R}^{\lvert k \rvert \times n}$	$e_1 \cdot r \cdot e_2$
ComplEx	$\mathrm{E} : \mathbb{C}^{\lvert E \rvert \times n}$	$\mathbf{W} : \mathbb{C}^{\lvert k \rvert \times n}$	$\mathrm{Re}\langle e_1, r, \bar{e}_2 \rangle$

Антисимметричность

Теперь можно

Trouillon et al. Complex Embeddings for Simple Link Prediction. ICML 2016

KGE - ComplEx & Patterns

Tensor Factorization

$$\mathrm{Re}\langle e_1, r, \bar{e}_2 \rangle$$

Симметричность

$$\mathrm{Re}\langle e_1, r, \bar{e}_2 \rangle = \mathrm{Re}\langle e_2, r, \bar{e}_1 \rangle$$

При Im(*r*) = 0

Инверсия

Возможно при $r_1 = \bar{r}_2$

Антисимметричность

$$\mathrm{Re}\langle e_1, r_1, \bar{e}_2 \rangle \neq \mathrm{Re}\langle e_2, r_2, \bar{e}_1 \rangle$$

Композиция

Отношения 1-N

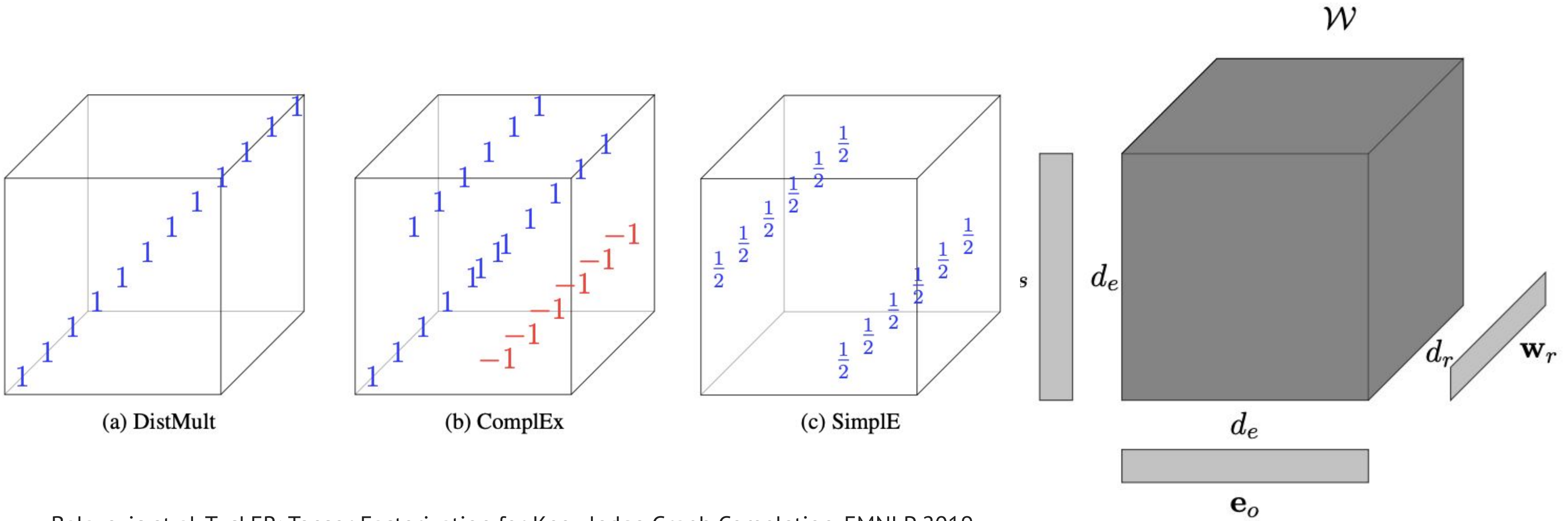
$$\mathrm{Re}\langle e_1, r, \bar{e}_2 \rangle \neq \mathrm{Re}\langle e_1, r, \bar{e}_n \rangle$$

KGE - TuckER

Tensor Factorization

Goal - factorize a sparse 3D tensor to dense core W, entities E and relations R

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



Balazevic et al. TuckER: Tensor Factorization for Knowledge Graph Completion. EMNLP 2019

KGE - TransE

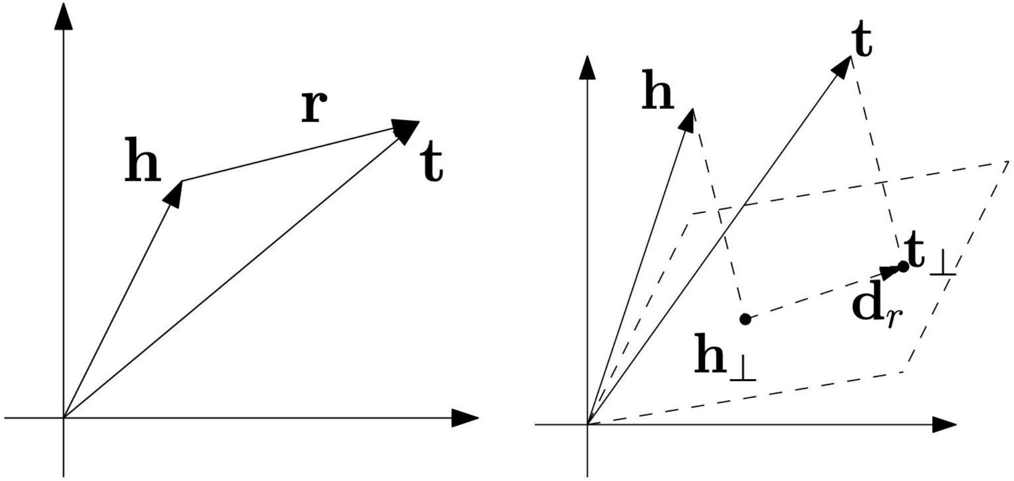
Tensor Factorization

Translation

Translate entities and relations into one embedding space

$$h + r \approx t \quad \text{Moscow} + \text{capitalOf} \approx \text{Russia}$$

$$||h||_2^2 = ||t||_2^2 = 1$$



(a) TransE (b) TransH

Bordes et al. Translating Embeddings for Modeling Multi-relational Data. NIPS 2013
Wang et al. Knowledge Graph Embedding by Translating on Hyperplanes. AAAI 2014

KGE - TransE

Tensor Factorization

Translation

Translate entities and relations into one embedding space

Algorithm 1 Learning TransE

input Training set $S = \{(h, \ell, t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\boldsymbol{\ell} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$

$\quad \boldsymbol{\ell} \leftarrow \boldsymbol{\ell} / \|\boldsymbol{\ell}\|$ for each $\ell \in L$

$\quad \mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

loop

$\quad \mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$

$\quad S_{batch} \leftarrow \text{sample}(S, b)$ // sample a minibatch of size b

$\quad T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets

$\quad$ **for** $(h, \ell, t) \in S_{batch}$ **do**

$\qquad (h', \ell, t') \leftarrow \text{sample}(S'_{(h,\ell,t)})$ // sample a corrupted triplet

$\qquad T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell, t'))\}$

$\quad$ **end for**

$\quad$ Update embeddings w.r.t. $\displaystyle\sum_{((h,\ell,t),(h',\ell,t')) \in T_{batch}} \nabla \left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+$

end loop

Bordes et al. Translating Embeddings for Modeling Multi-relational Data. NIPS 2013

KGE - TransE & Patterns

Tensor Factorization

$$||h + r - t||$$

Translation

Симметричность

$$||h + r - t|| \neq ||t + r - h||$$

Инверсия

$$h + r_1 \approx t$$
$$t + r_2 \approx h \rightarrow r_2 = -r_1$$

Антисимметричность

$$h + r_1 \approx t$$
$$t + r_2 \approx h \rightarrow r_2 \neq -r_1$$

Отношения 1-N

$$h + r \approx t_1,$$
$$h + r \approx t_n$$
$$\rightarrow t_1 = t_n$$

Композиция

$$h + r_1 \approx t_1,$$
$$t_1 + r_2 \approx t_2$$
$$\rightarrow h + (r_1 + r_2) \approx t_2$$

Bordes et al. Translating Embeddings for Modeling Multi-relational Data. NIPS 2013

KGE - TransE

Tensor Factorization

Translation

LOTS of models

TABLE 9
Knowledge graph embedding using margin-based ranking loss.

GE Algorithm	Energy Function $\mathbf{f_r(h,t)}$
TransE [91]	$\|h+r-t\|_{l1}$
TKRL [53]	$\|M_{rh}h+r-M_{rt}t\|$
TransR [15]	$\|hM_r+r-tM_r\|_2^2$
CTransR [15]	$\|hM_r+r_c-tM_r\|_2^2+\alpha\|r_c-r\|_2^2$
TransH [14]	$\|(h-w_r^T hw_r)+d_r-(t-w_r^T tw_r)\|_2^2$
SePLi [39]	$\frac{1}{2}\|W_ie_{ih}+b_i-e_{it}\|^2$
TransD [125]	$\|M_{rh}h+r-M_{rt}t\|_2^2$
TranSparse [126]	$\|M_r^h(\theta_r^h)h+r-M_r^t(\theta_r^t)t\|_{l_1/2}^2$
m-TransH [127]	$\|\sum_{\rho\in\mathcal{M}(R_r)} a_r(\rho)\mathbb{P}_{n_r}(t(\rho))+b_r\|^2, t\in\mathcal{N}^{\mathcal{M}(R_r)}$
DKRL [128]	$\|h_d+r-t_d\|+\|h_d+r-t_s\|+\|h_s+r-t_d\|$
ManifoldE [129]	Sphere: $\|\varphi(h)+\varphi(r)-\varphi(t)\|^2$ Hyperplane: $(\varphi(h)+\varphi(r_{head}))^T(\varphi(t)+\varphi(r_{tail}))$ φ is the mapping function to Hilbert space
TransA [130]	$\|h+r-t\|$
puTransE [43]	$\|h+r-t\|$
KGE-LDA [60]	$\|h+r-t\|_{l1}$
SE [90]	$\|R_uh-R_ut\|_{l1}$
SME [92] linear SME [92] bilinear	$(W_{u1}r+W_{u2}h+b_u)^T(W_{v1}r+W_{v2}t+b_v)$ $(W_{u1}r+W_{u2}h+b_u)^T(W_{v1}r+W_{v2}t+b_v)$
SSP [59]	$-\lambda\|e-s^Tes\|_2^2+\|e\|_2^2, S(s_h,s_t)=\frac{s_h+s_t}{\|s_h+s_t\|_2^2}$
NTN [131]	$u_r^T\tanh(h^TW_rt+W_{rh}h+W_{rt}t+b_r)$
HOLE [132]	$r^T(h\star t)$, where $\star$ is circular correlation
MTransE [133]	$\|h+r-t\|_{l1}$

Cai et al. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications. IEEE TKDE 2017

KGE - Incorporating OWL Rules

Tensor Factorization

Translation

$$\min_{\theta} \sum_{(h,r,t)\in\mathcal{S}} \alpha^r_{h,t} \log(1+\exp(-y^r_{h,t} f^r_{h,t})) + \lambda \sum_{i=1}^{l} \frac{\mathcal{R}_i}{N_i}$$
$$\text{subject to} \quad \|h\| = 1 \text{ and } \|t\| = 1 .$$

Rule	Definition $\forall \mathtt{h}, \mathtt{t}, \mathtt{s} \in \mathcal{E} : \ldots$	Formulation based on score function	Formulation based on NN	Equivalent regularization form (Denoted as $\mathcal{R}_i$ in Equation (2))
Equivalence	$(\mathtt{h}, r_1, \mathtt{t}) \Leftrightarrow (\mathtt{h}, r_2, \mathtt{t})$	$f^{r_1}_{h,t} = f^{r_2}_{h,t} + \xi_{h,t}$	$\Phi^T_{h,t}(\boldsymbol{\beta}^{r_1} - \boldsymbol{\beta}^{r_2}) = \xi_{h,t}$	$\max(\|\boldsymbol{\beta}^{r_1} - \boldsymbol{\beta}^{r_2}\|_1 - \xi_{\text{Eq}}, 0)$
Symmetric	$(\mathtt{h}, r, \mathtt{t}) \Leftrightarrow (\mathtt{t}, r, \mathtt{h})$	$f^r_{h,t} = f^r_{t,h} + \xi_{h,t}$	$(\Phi_{h,t} - \Phi_{t,h})^T \boldsymbol{\beta}^r = \xi_{h,t}$	$\max(\lvert(\Phi_{h,t} - \Phi_{t,h})^T \boldsymbol{\beta}^r\rvert - \xi_{\text{Sy}}, 0)$
Asymmetric	$(\mathtt{h}, r, \mathtt{t}) \Rightarrow \neg(\mathtt{t}, r, \mathtt{h})$	$f^r_{h,t} = f^r_{t,h} + \mathcal{M}_{h,t}$	$(\Phi_{h,t} - \Phi_{t,h})^T \boldsymbol{\beta}^r = \mathcal{M}$	NC
Negation	$(\mathtt{h}, r_1, \mathtt{t}) \Leftrightarrow \neg(\mathtt{h}, r_2, \mathtt{t})$	$f^{r_1}_{h,t} = \mathcal{M} - f^{r_2}_{h,t} + \xi_{h,t}$	$\Phi^T_{h,t}(\boldsymbol{\beta}^{r_1} + \boldsymbol{\beta}^{r_2}) = \mathcal{M} + \xi_{h,t}$	NC
Implication	$(\mathtt{h}, r_1, \mathtt{t}) \Rightarrow (\mathtt{h}, r_2, \mathtt{t})$	$f^{r_1}_{h,t} \le f^{r_2}_{h,t}$	$\Phi^T_{h,t}(\boldsymbol{\beta}^{r_1} - \boldsymbol{\beta}^{r_2}) \le 0$	$\max(\sum_i (\boldsymbol{\beta}^{r_1}_i - \boldsymbol{\beta}^{r_2}_i) + \xi_{\text{Im}}, 0)$
Inverse	$(\mathtt{h}, r_1, \mathtt{t}) \Rightarrow (\mathtt{t}, r_2, \mathtt{h})$	$f^{r_1}_{h,t} \le f^{r_2}_{t,h}$	$\Phi^T_{h,t}\boldsymbol{\beta}^{r_1} - \Phi^T_{t,h}\boldsymbol{\beta}^{r_2} \le 0$	$\max(\Phi^T_{h,t}\boldsymbol{\beta}^{r_1} - \Phi^T_{t,h}\boldsymbol{\beta}^{r_2} + \xi_{\text{In}}, 0)$
Reflexivity	$(\mathtt{h}, r, \mathtt{h})$	$f^r_{h,h} = \mathcal{M} - \xi_{h,h}$	$\Phi^T_{h,h}\boldsymbol{\beta}^r = \mathcal{M} - \xi_{h,h}$	NC
Irreflexive	$\neg(\mathtt{h}, r, \mathtt{h})$	$f^r_{h,h} = \xi_{h,h}$	$\Phi^T_{h,h}\boldsymbol{\beta}^r = \xi_{h,h}$	NC
Transitivity	$(\mathtt{h}, r, \mathtt{t}) \wedge (\mathtt{t}, r, \mathtt{s}) \Rightarrow (\mathtt{h}, r, \mathtt{s})$	$\sigma(f^r_{h,s}) \ge \sigma(f^r_{h,t}) \times \sigma(f^r_{t,s})$	$\sigma(\Phi_{h,t}\boldsymbol{\beta}^r) \times \sigma(\Phi_{t,s}\boldsymbol{\beta}^r) - \sigma(\Phi^T_{h,s}\boldsymbol{\beta}^r) \le 0$	$\max(\sigma(\Phi_{h,t}\boldsymbol{\beta}^r) \times \sigma(\Phi_{t,s}\boldsymbol{\beta}^r) - \sigma(\Phi^T_{h,s}\boldsymbol{\beta}^r) + \xi_{\text{Tr}}, 0)$
Composition	$(\mathtt{h}, r_1, \mathtt{t}) \wedge (\mathtt{t}, r_2, \mathtt{s}) \Rightarrow (\mathtt{h}, r_3, \mathtt{s})$	$\sigma(f^{r_1}_{h,s}) \ge \sigma(f^{r_2}_{h,t}) \times \sigma(f^{r_3}_{t,s})$	$\sigma(\Phi_{h,t}\boldsymbol{\beta}^{r_1}) \times \sigma(\Phi_{t,s}\boldsymbol{\beta}^{r_2}) - \sigma(\Phi^T_{h,s}\boldsymbol{\beta}^{r_3}) \le 0$	$\max(\sigma(\Phi_{h,t}\boldsymbol{\beta}^{r_1}) \times \sigma(\Phi_{t,s}\boldsymbol{\beta}^{r_2}) - \sigma(\Phi^T_{h,s}\boldsymbol{\beta}^{r_3}) + \xi_{\text{Co}}, 0)$

Table 1: Formulation and representation of rules (NC: Not considered for implementation).

Nayyeri et al. LogicENN: A Neural Based Knowledge Graphs Embedding Model with Logical Rules. https://arxiv.org/abs/1908.07141

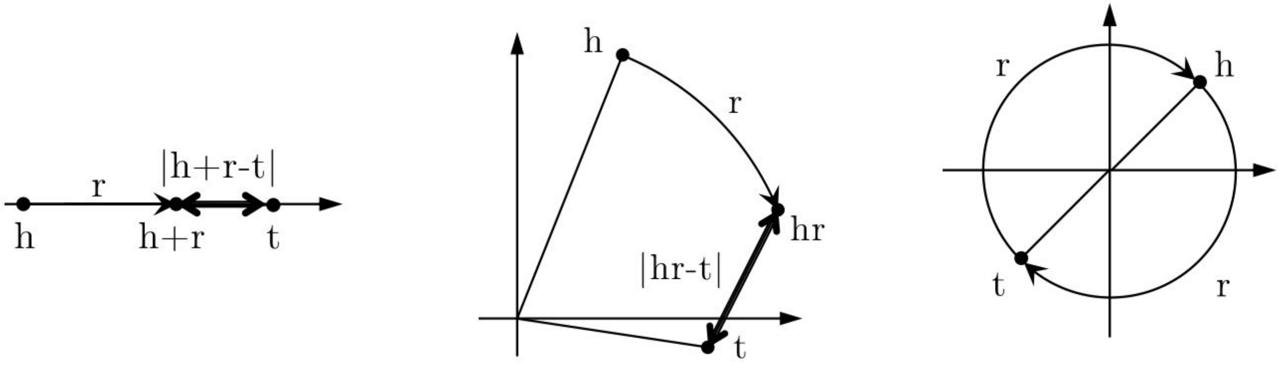
KGE - RotatE

Tensor Factorization

Translation

Idea:
Entities are vectors in **complex space**

Relations: rotations in **complex space**



(a) TransE models $\mathbf{r}$ as translation in real line.

(b) RotatE models $\mathbf{r}$ as rotation in complex plane.

(c) RotatE: an example of modeling symmetric relations $\mathbf{r}$ with $r_i = -1$

Figure 1: Illustrations of TransE and RotatE with only 1 dimension of embedding.

Score function: $d_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|$ $\quad |r_i| = 1$

Loss & Optimization: $L = -\log \sigma(\gamma - d_r(\mathbf{h}, \mathbf{t})) - \sum_{i=1}^{n} \frac{1}{k} \log \sigma(d_r(\mathbf{h}'_i, \mathbf{t}'_i) - \gamma),$

Sun et al. RotatE: Knowledge Graph Embedding By Relational Rotation In Complex Space. ICLR 2019

KGE - RotatE & Patterns

Translation

Model	Score Function	Symmetry	Antisymmetry	Inversion	Composition
SE	$-\lVert \boldsymbol{W}_{r,1}\mathbf{h} - \boldsymbol{W}_{r,2}\mathbf{t} \rVert$	✗	✗	✗	✗
TransE	$-\lVert \mathbf{h} + \mathbf{r} - \mathbf{t} \rVert$	✗	✓	✓	✓
TransX	$-\lVert g_{r,1}(\mathbf{h}) + \mathbf{r} - g_{r,2}(\mathbf{t}) \rVert$	✓	✓	✗	✗
DistMult	$\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$	✓	✗	✗	✗
ComplEx	$\mathrm{Re}(\langle \mathbf{h}, \mathbf{r}, \overline{\mathbf{t}} \rangle)$	✓	✓	✓	✗
RotatE	$-\lVert \mathbf{h} \circ \mathbf{r} - \mathbf{t} \rVert$	✓	✓	✓	✓

Table 2: The pattern modeling and inference abilities of several models.

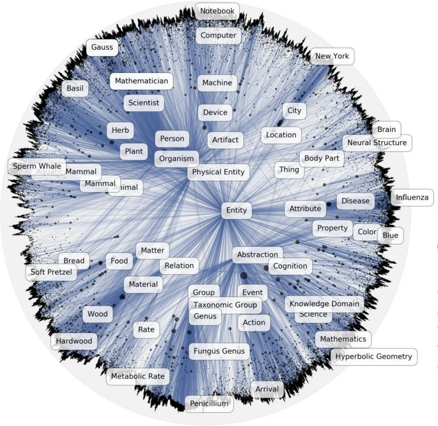
Sun et al. Rotate: Knowledge graph embedding by relational rotation in complex space. ICLR 2019

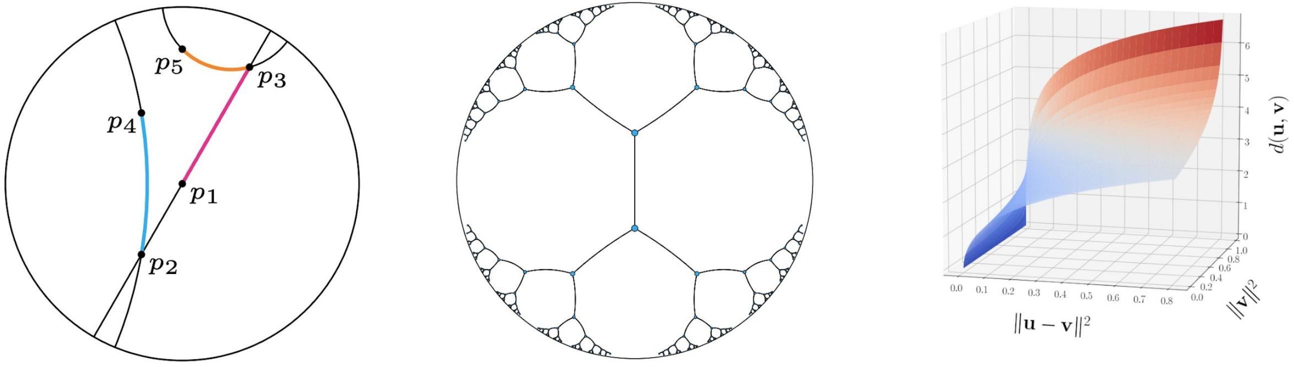
KGE - Hyperbolic

Tensor Factorization

Translation

Goal: embed hierarchical structures into a hyperbolic manifold.





(a) Geodesics of the Poincaré disk (b) Embedding of a tree in $\mathcal{B}^2$ (c) Growth of Poincaré distance

Figure 1: (a) Due to the negative curvature of $\mathcal{B}$, the distance of points increases exponentially (relative to their Euclidean distance) the closer they are to the boundary. (c) Growth of the Poincaré distance $d(\boldsymbol{u}, \boldsymbol{v})$ relative to the Euclidean distance and the norm of $\boldsymbol{v}$ (for fixed $\|\boldsymbol{u}\| = 0.9$). (b) Embedding of a regular tree in $\mathcal{B}^2$ such that all connected nodes are spaced equally far apart (i.e., all black line segments have identical hyperbolic length).

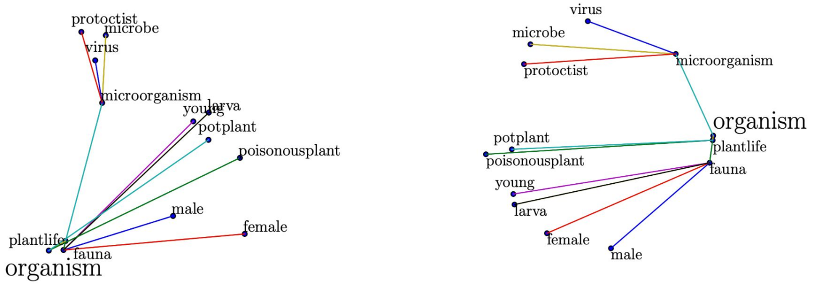
Nickel et al. Poincaré Embeddings for Learning Hierarchical Representations. NIPS 2017

KGE - Hyperbolic

Tensor Factorization

Translation

Goal: embed hierarchical structures into a hyperbolic manifold



(a) ROTE embeddings. (b) ROTH embeddings.

Figure 6: Visualizations of the embeddings learned by ROTE and ROTH on a sub-tree of WN18RR for the *hypernym* relation. In contrast to ROTE, ROTH preserves hierarchies by learning tree-like embeddings.

- ✓ **Хорошо работает на иерархических графах**
- ✓ **Эффективны на малых размерностях (32-64d)**

Chami et al. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. ACL 2020

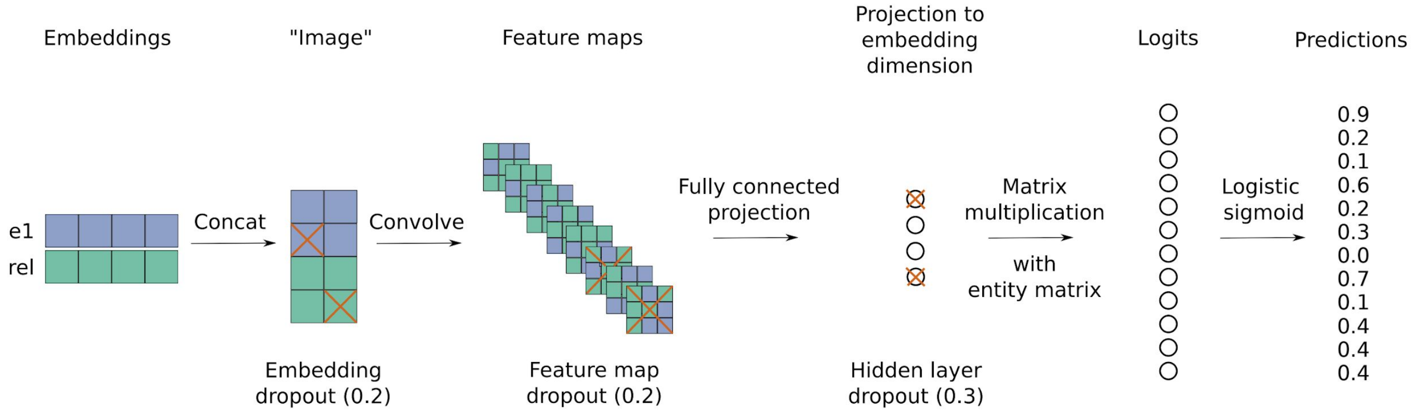
KGE - ConvE

Tensor Factorization

Translation

Convolution

Goal: CNNs for predicting a probability of the object



Score function: $\psi_r(\mathbf{e}_s, \mathbf{e}_o) = f(\text{vec}(f([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_r}] * \omega))\mathbf{W})\mathbf{e}_o,$

Loss & Optimization: $\mathcal{L}(p, t) = -\frac{1}{N} \sum_i (t_i \cdot \log(p_i) + (1 - t_i) \cdot \log(1 - p_i)),$

Minervini et al. Convolutional 2D Knowledge Graph Embeddings. AAAI 2018

KGE - ConvKB

Tensor Factorization

Translation

Convolution

Score function:

$$f(h, r, t) = \text{concat}\left(g\left([\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t] * \boldsymbol{\Omega}\right)\right) \cdot \mathbf{w}$$

Loss & Optimization:

$$\mathcal{L} = \sum_{(h,r,t) \in \{\mathcal{G} \cup \mathcal{G}'\}} \log\left(1 + \exp\left(l_{(h,r,t)} \cdot f(h, r, t)\right)\right) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

$$\text{in which, } l_{(h,r,t)} = \begin{cases} 1 \text{ for } (h, r, t) \in \mathcal{G} \\ -1 \text{ for } (h, r, t) \in \mathcal{G}' \end{cases}$$

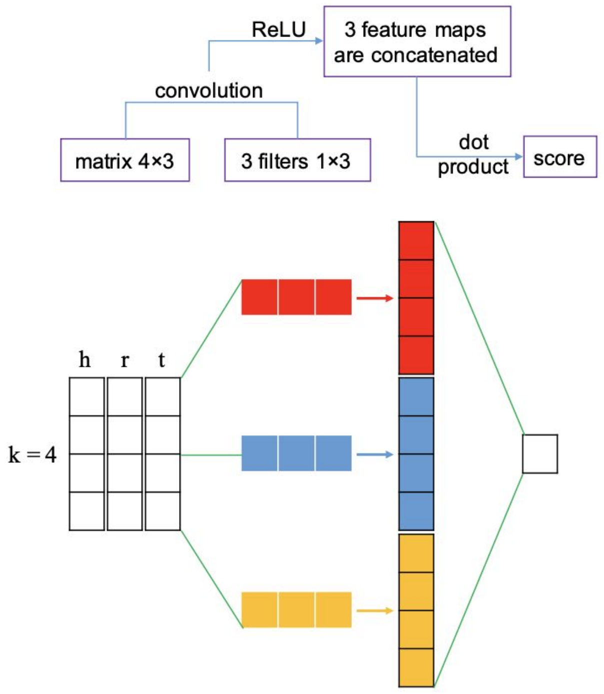


Figure 1: Process involved in ConvKB (with the embedding size $k = 4$, the number of filters $\tau = 3$ and the activation function $g = \text{ReLU}$ for illustration purpose).

Nguyen et al. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. NAACL 2018

KGE - CoKE

Tensor Factorization

Translation

Transformer

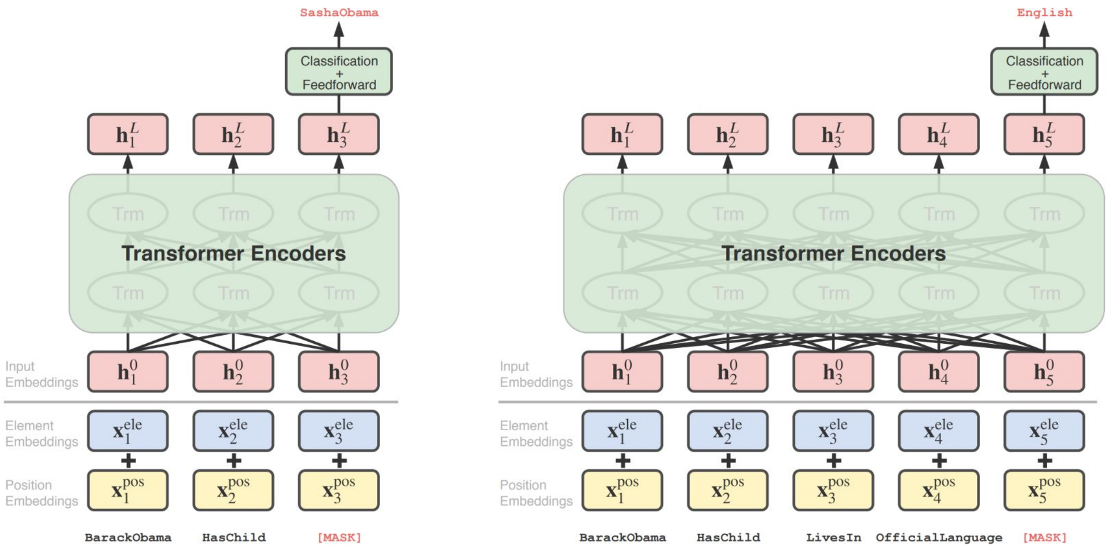
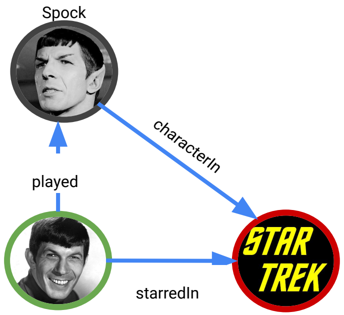


Figure 2: Overall framework of CoKE. An edge (left) or a path (right) is given as an input sequence, with an entity replaced by a special token [MASK]. The input is then fed into a stack of Transformer encoder blocks. The final hidden state corresponding to [MASK] is used to predict the target entity.

Wang et al. CoKE: Contextualized Knowledge Graph Embeddings. arxiv 2019

Training & Evaluation

KGE - Training



Entity matrix

$$\mathbf{E} : \mathbb{R}^{|E| \times n}$$

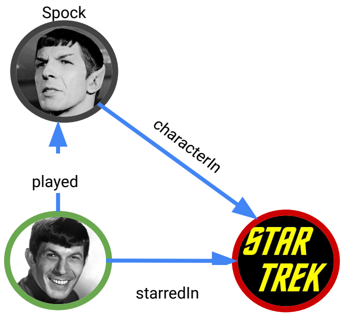
Spock = [0.1, 0.2, 0.3]
Leonard Nimoy = [0.4, 0.8, 0.1]
Star Trek = [0.22, 0.34, 0.87]

Relations matrix

$$\mathbf{W} : \mathbb{R}^{|k| \times n}$$

characterIn = [0.1, 0.1, 0.6]
played = [0.2, 0.3, 0.4]
starredIn = [0.9, -0.2, 0.1]

KGE - Training



Optimization

Loss Function

Negative Sampling

Entity matrix

$$\mathrm{E} : \mathbb{R}^{|E| \times n}$$

Spock = [0.1, 0.2, 0.3]
Leonard Nimoy = [0.4, 0.8, 0.1]
Star Trek = [0.22, 0.34, 0.87]

Relations matrix

$$\mathrm{W} : \mathbb{R}^{|k| \times n}$$

characterIn = [0.1, 0.1, 0.6]
played = [0.2, 0.3, 0.4]
starredIn = [0.9, -0.2, 0.1]

KGE - Training - sLCWA vs LCWA

Local Closed World Assumption (LCWA)

- Предсказываем распределение по всем сущностям на выходе (1-N scoring)
- Classification losses:
  - BCE, CE
- Часто добавляют inverse relations

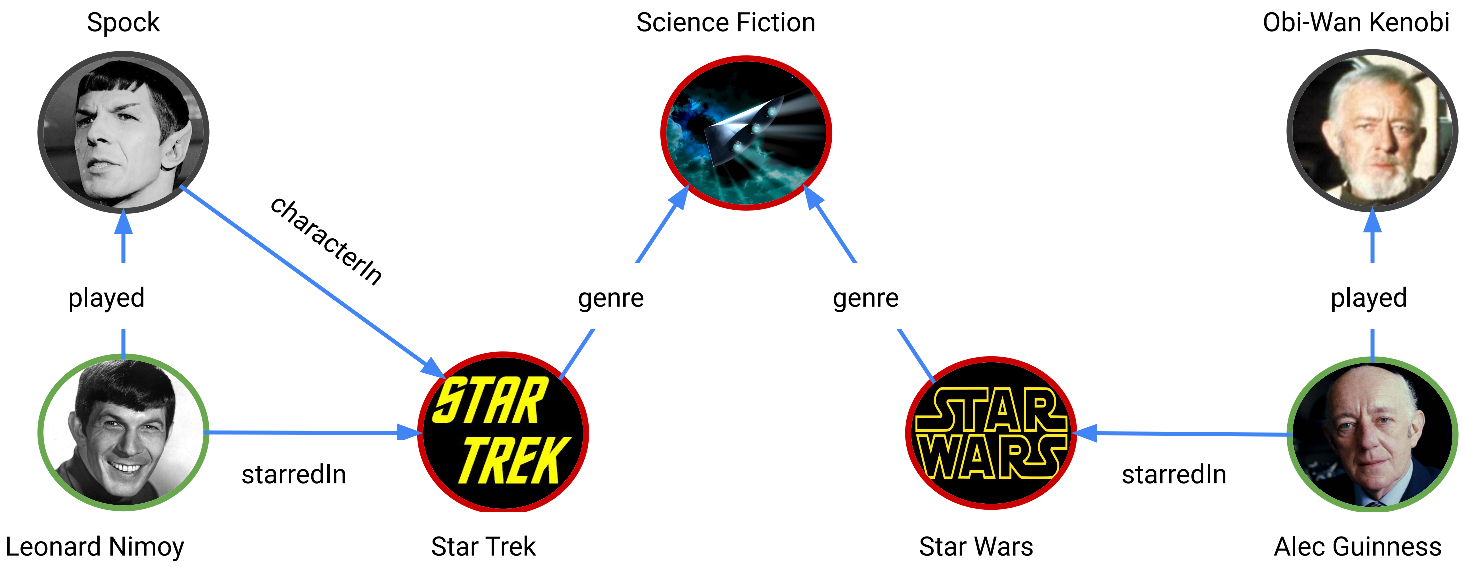
stochastic Local Closed World Assumption (sLCWA)

- Negative sampling

$$f(h, r, t) > f(h, r, t')$$

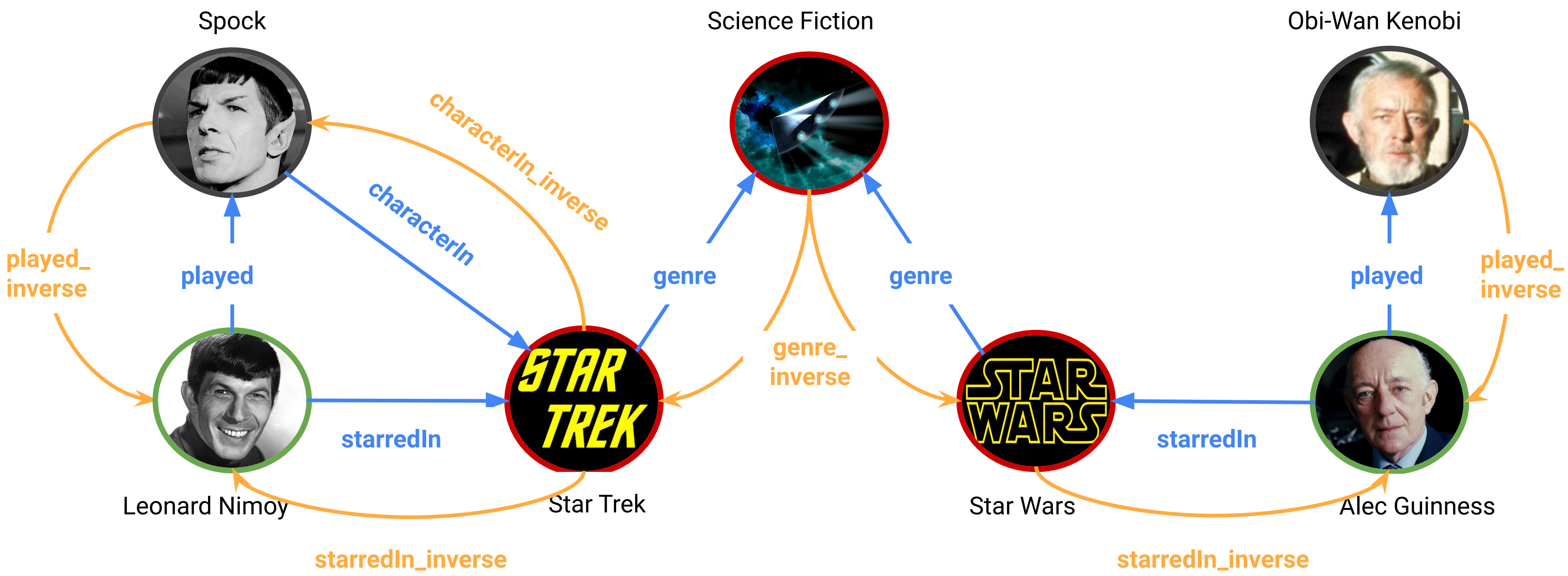
- Contrastive losses:
  - Margin Ranking Loss
  - Self-Adversarial Loss
  - Softplus Loss

KGE - Adding Inverse Relations



Nickel et al. A review of relational machine learning for knowledge graphs. IEEE. 2015

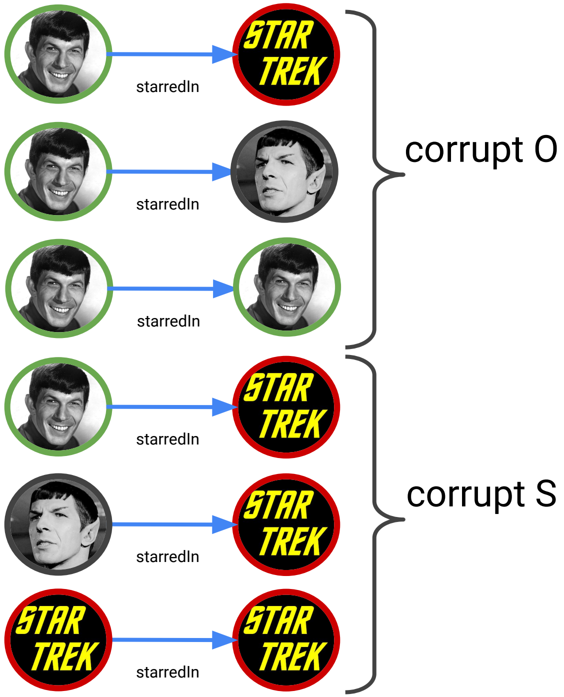
KGE - Adding Inverse Relations



- ➢ 2x больше триплетов
- ➢ 2x больше типов предикатов

Nickel et al. A review of relational machine learning for knowledge graphs. IEEE. 2015

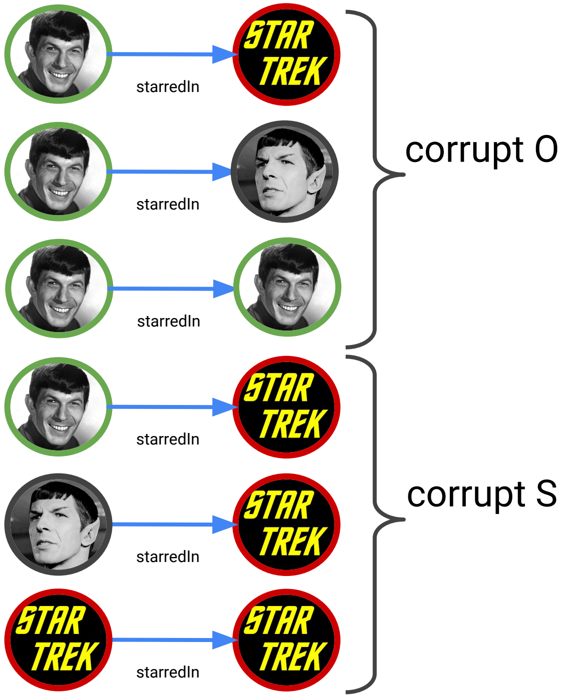
KGE - Training - sLCWA + Margin Loss



$$L(\Omega) = \sum_{(e_1,r,e_2)\in T} \sum_{(e'_1,r,e'_2)\in T'} \max\{S_{(e'_1,r,e'_2)} - S_{(e_1,r,e_2)} + 1, 0\}$$

Negative sampling: incorrect triples should have lower (higher) score than correct triples

KGE - Training - sLCWA + Margin Loss



$$L(\Omega) = \sum_{(e_1,r,e_2)\in T} \sum_{(e'_1,r,e'_2)\in T'} \max\{S_{(e'_1,r,e'_2)} - S_{(e_1,r,e_2)} + 1, 0\}$$

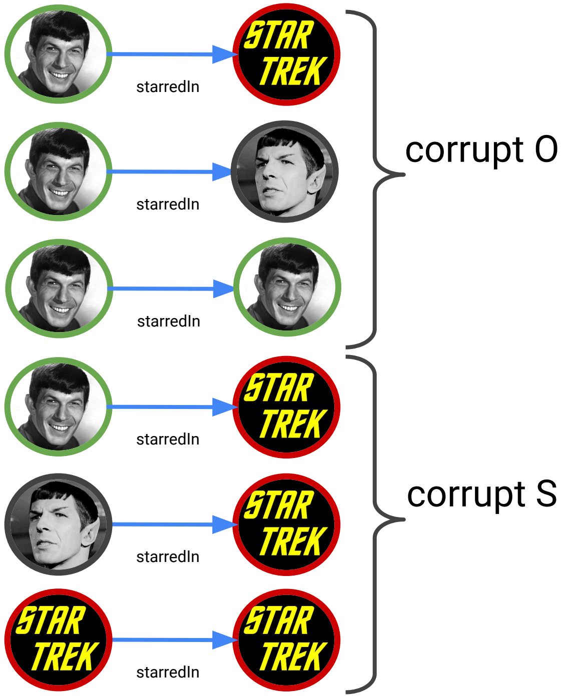
Negative sampling: incorrect triples should have lower (higher) score than correct triples

Negative Sampling Self-Adversarial Loss (NSSAL) (Sun et al)

$$L = -\log \sigma(\gamma - d_r(\mathbf{h},\mathbf{t})) - \sum_{i=1}^{n} p(h'_i, r, t'_i) \log \sigma(d_r(\mathbf{h}'_i, \mathbf{t}'_i) - \gamma)$$

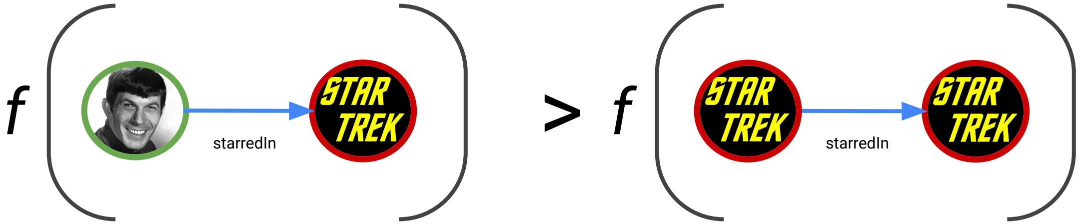
$$p(h'_j, r, t'_j | \{(h_i, r_i, t_i)\}) = \frac{\exp \alpha f_r(\mathbf{h}'_j, \mathbf{t}'_j)}{\sum_i \exp \alpha f_r(\mathbf{h}'_i, \mathbf{t}'_i)}$$

KGE - Training - sLCWA + Margin Loss



$$L(\Omega) = \sum_{(e_1,r,e_2)\in T} \sum_{(e'_1,r,e'_2)\in T'} \max\{S_{(e'_1,r,e'_2)} - S_{(e_1,r,e_2)} + 1, 0\}$$

Negative sampling: incorrect triples should have lower (higher) score than correct triples



KGE - Training - LCWA + (Binary) Cross-Entropy Loss

$$L = -\frac{1}{n_e}\sum_{i=1}^{n_e}(\mathbf{y}^{(i)}\log(\mathbf{p}^{(i)}) + (1-\mathbf{y}^{(i)})\log(1-\mathbf{p}^{(i)})),$$

Model's output is usually sigmoid / log softmax

$$\sigma(score(e_1, r, e_2))$$

KGE - Training - sLCWA vs LCWA

Local Closed World Assumption (LCWA)

- \+ Быстрее сходится
- \+ Быстрый evaluation

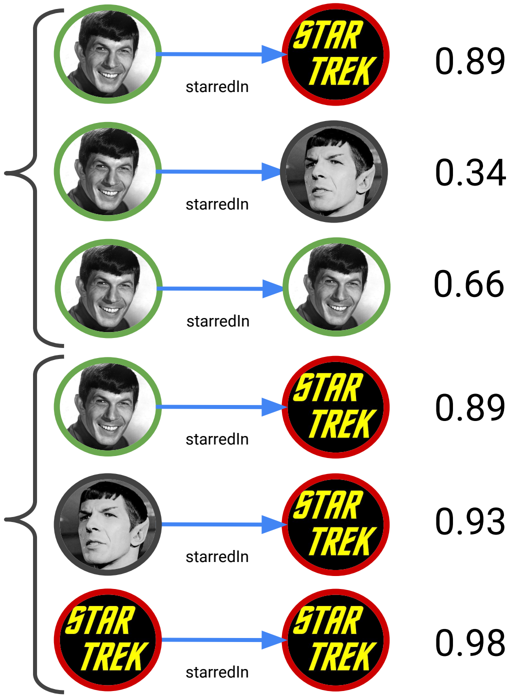
- \- Output shape: ***[bs, num_entities]***
- \- Батчи съедают много GPU памяти
- \- Софт-лимит: 100K сущностей

stochastic Local Closed World Assumption (sLCWA)

- \+ Output shape: ***[bs*num_negs, 1]***
- \+ Работает на больших графах
- \+ Меньше GPU consumption

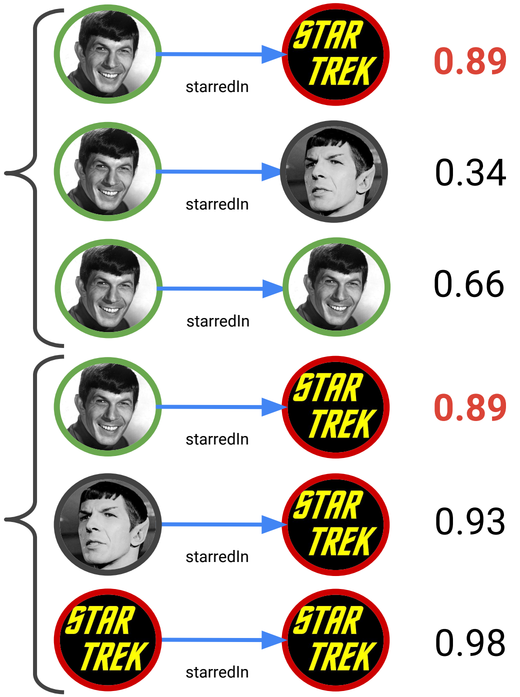
- \- Медленно сходится
- \- Нужно дополнительно подбирать margin , temperature
- \- Долгий evaluation

KGE - Training - Metrics



$$MRR = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{rank_i}$$

KGE - Training - Metrics

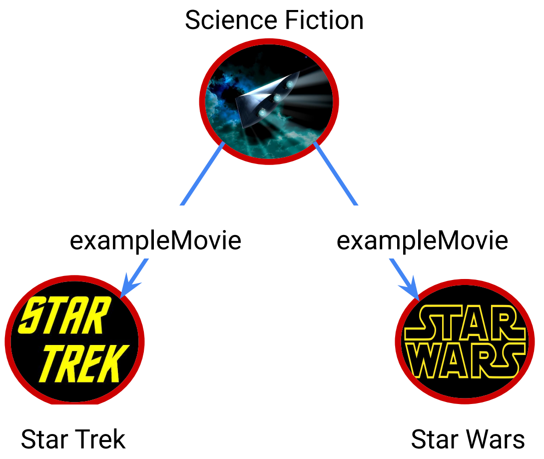


$$MRR = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{rank_i}$$

Как правило, все метрики подсчитываются в **отфильтрованном** режиме

	Corrupt O	Corrupt S	MR	MRR	Avg H@k
Rank	1	3	2	0.66	
Reciprocal rank	1	⅓			
Hits@1	1	0			0.5
Hits@3	1	1			1.0
Hits@10	1	1			1.0

KGE - Training - Filtered Metrics



- Часто, у пары (head, relation) может быть несколько корректных объектов (tails).
- Необходима поправка в ранжирование - **отфильтровать** прочие корректные ответы
- Допустим, оцениваем предсказания триплета **(Science Fiction, exampleMovie, Star Wars)**

rank = 2	Unfiltered	0.03	0.87	0.18	0.85	0.10	0.23	0.13
rank = 1	Filtered	0.03	*-inf*	0.18	0.85	0.10	0.23	0.13

Datasets & Benchmarks

Datasets

	WN18	FB15k	FB15k-237 (Freebase)	WN18RR (WordNet)	CoDEx (Wikidata)	YAGO 3-10	OGB Wiki KG	KDD Cup Wikidata
# entities	Don’t use (please)		15K	40K	2-70K	125K	2.5M	87M
# edges			272k	80k	33-550K	1M	13M	504M
# relations			237	11	42-69	34	~1000	1,315

… и много других

KGE - Benchmarking / SOTA

Tensor Factorization

Принято демонстрировать SOTA по сравнению с baselines

		WN18RR				FB15k-237			
	Linear	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
DistMult (Yang et al., 2015)	yes	.430	.490	.440	.390	.241	.419	.263	.155
ComplEx (Trouillon et al., 2016)	yes	.440	.510	.460	.410	.247	.428	.275	.158
Neural LP (Yang et al., 2017)	no	–	–	–	–	.250	.408	–	–
R-GCN (Schlichtkrull et al., 2018)	no	–	–	–	–	.248	.417	.264	.151
MINERVA (Das et al., 2018)	no	–	–	–	–	–	.456	–	–
ConvE (Dettmers et al., 2018)	no	.430	.520	.440	.400	.325	.501	.356	.237
HypER (Balažević et al., 2019)	no	.465	.522	.477	.436	.341	.520	.376	.252
M-Walk (Shen et al., 2018)	no	.437	–	.445	.414	–	–	–	–
RotatE (Sun et al., 2019)	no	–	–	–	–	.297	.480	.328	.205
TuckER (ours)	yes	**.470**	**.526**	**.482**	**.443**	**.358**	**.544**	**.394**	**.266**

Balazevic et al. TuckER: Tensor Factorization for Knowledge Graph Completion. EMNLP 2019

KGE - Benchmarking / SOTA

Tensor Factorization

Translation

Принято демонстрировать SOTA по сравнению с baselines

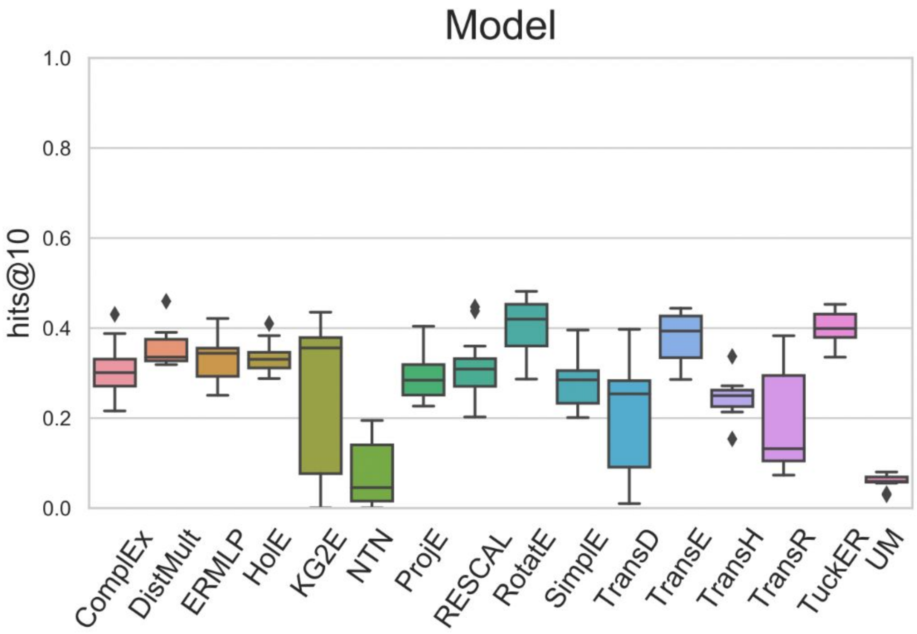
	FB15k-237					**WN18RR**				
	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE [♥]	357	.294	-	-	.465	3384	.226	-	-	.501
DistMult	254	.241	.155	.263	.419	5110	.43	.39	.44	.49
ComplEx	339	.247	.158	.275	.428	5261	.44	.41	.46	.51
ConvE	244	.325	.237	.356	.501	4187	.43	.40	.44	.52
pRotatE	178	.328	.230	.365	.524	**2923**	.462	.417	.479	.552
RotatE	**177**	**.338**	**.241**	**.375**	**.533**	3340	**.476**	**.428**	**.492**	**.571**

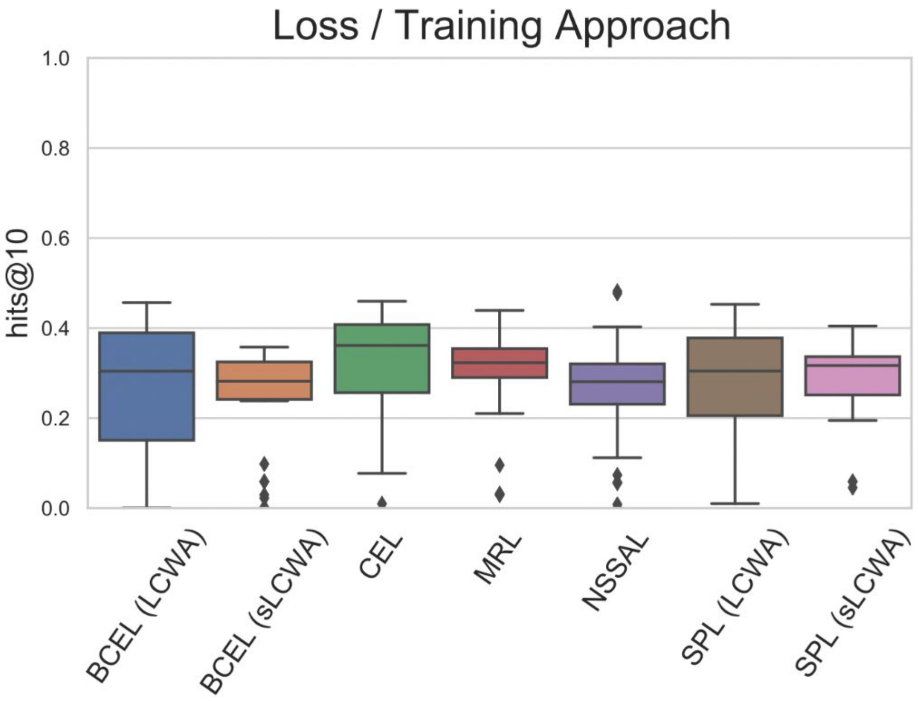
Table 5: Results of several models evaluated on the FB15k-237 and WN18RR datasets. Results of [♥] are taken from (Nguyen et al., 2017). Other results are taken from (Dettmers et al., 2017).

Sun et al. RotatE: Knowledge Graph Embedding By Relational Rotation In Complex Space. ICLR 2019

KGE - Benchmarking / SOTA

Baselines strike back - правильный подбор гиперпараметров делает старые модели сильными





Dataset: FB15k-237

Ali et al. Bringing Light Into the Dark: A Large-scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework. arxiv:2006.13365

KG Embeddings Library: PyKEEN

→ **PyTorch** 😍
→ 26 datasets + your own graphs
→ 28 KG embedding models and counting

→ 7 losses
→ 6 optimizers
→ 16 metrics
→ 5 regularizers
→ 2 training loops
→ 3 negative samplers
→ Tracking in MLFlow, WANDB, TensorBoard, and more

📈 Benchmarked!

Ali et al. Bringing Light Into the Dark: A Large-scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework. arxiv:2006.13365

PyKEEN

build passing | License MIT | DOI 10.5281/zenodo.3982977 | Optuna integrated

https://github.com/pykeen/pykeen

В следующей серии

1. Introduction
2. Представление знаний в графах - RDF & RDFS & OWL
3. Хранение знаний в графах - SPARQL & Graph Databases
4. Однородность знаний - RDF* & Wikidata & SHACL & ShEx
5. Интеграция данных в графы знаний - Semantic Data Integration
6. Введение в теорию графов - Graph Theory Intro
7. Векторные представления графов - Knowledge Graph Embeddings
8. **Машинное обучение на графах - Graph Neural Networks & KGs**
9. Некоторые применения - Question Answering & Query Embedding