# Графы знаний

Knowledge Graph Embeddings

М. Галкин, Д. Муромцев

# Представление знаний - онтологическое


```
LeonardNimoy starredIn StarTrek;
             played    Spock.
Spock        characterIn  StarTrek .
```

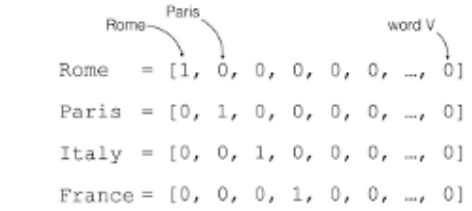
```
AlecGuinness starredIn StarWars;
             played    Obi-Wan.
StarWars     genre     SciFi .
```

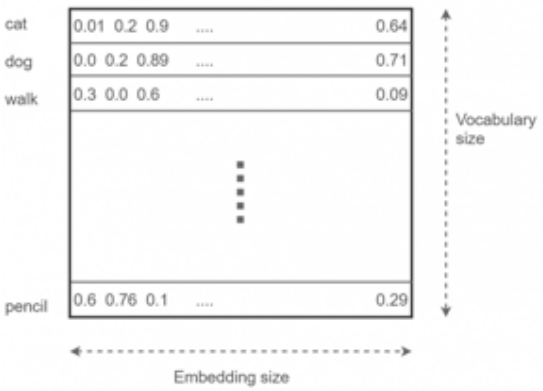


Так можно представить граф знаний с помощью онтологии и триплетов.

# Вспоминаем дистрибутивную семантику 2

- Самый просто способ представления – one-hot кодировка (каждое слово представлено в виде двоичного вектора длины n, где i-ое значение кодируется единицей на i-ой позиции и нулями на всех остальных)
- Более подробно можно посмотреть тут https://habr.com/ru/company/ods/blog/329410/
- Почему размер вектора отличается от размера словаря можно посмотреть тут https://neurohive.io/ru/osnovy-data-science/word2vec-vektornye-predstavlenija-slov-dlja-mashinnogo-obuchenija/





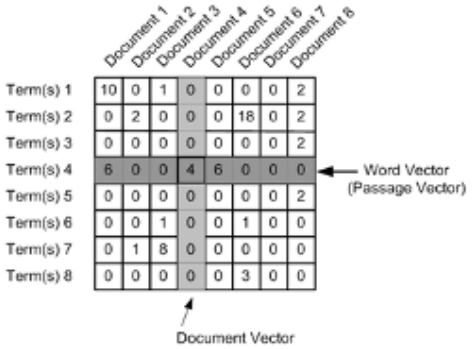

Так можно представить граф знаний с помощью онтологии и триплетов.

# Вспоминаем дистрибутивную семантику

| | *Работа* | *Деятельность* | *Труд* | *Действие* | *Дело* | *Создание* | *Творчество* |
|---|---|---|---|---|---|---|---|
| *Работа* | | 16 | 13 | 12 | 10 | 6 | 0 |
| *Деятельность* | 18 | | 9 | 12 | 5 | 6 | 2 |
| *Труд* | 37 | 7 | | 6 | 7 | 4 | 0 |
| *Действие* | 26 | 26 | 9 | | 6 | 8 | 2 |
| *Дело* | 39 | 11 | 22 | 27 | | 3 | 0 |
| *Создание* | 1 | 0 | 2 | 0 | 1 | | 3 |
| *Творчество* | 4 | 1 | 4 | 1 | 4 | 3 | |

- Пример для слов: *работа, действие, деятельность, труд, дело, создание, творчество*
- Сила семантических связей между этими словами определялась по словарным статьям толковых словарей

- если слова этой группы встречаются рядом в пределах словарных статей толковых и двуязычных словарей, то между словами регистрируются семантические связи
- чем чаще встречается та или иная пара слов вместе в словарных статьях, тем сильнее между словами этой пары
- Таким образом для каждого слова получается векторное представление, которое также можно вычислить применительно к различным документам из корпуса (с помощью TF-IDF)






Так можно представить граф знаний с помощью онтологии и триплетов.

# 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]

# Knowledge Graph Embeddings

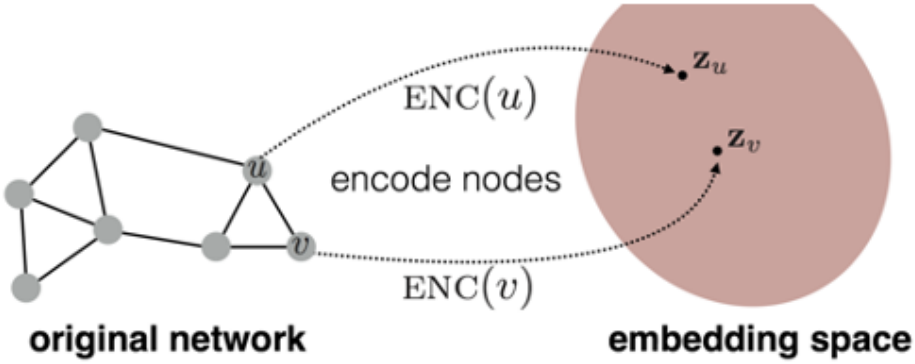
**Tensor Factorization**

**Translation**

**Convolution**

**Graph Neural Nets**

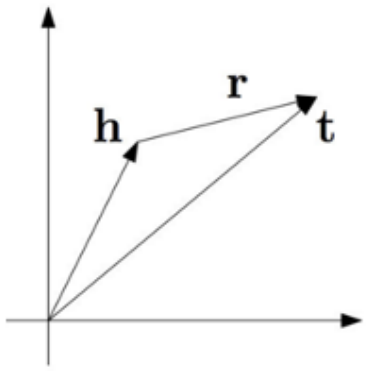
Поставим задачу: представим узлы графа так, чтобы их близость в векторном пространстве приблизительно соответствовало сходству в исходном графе




**original network** **embedding space**

# Knowledge Graph Embeddings

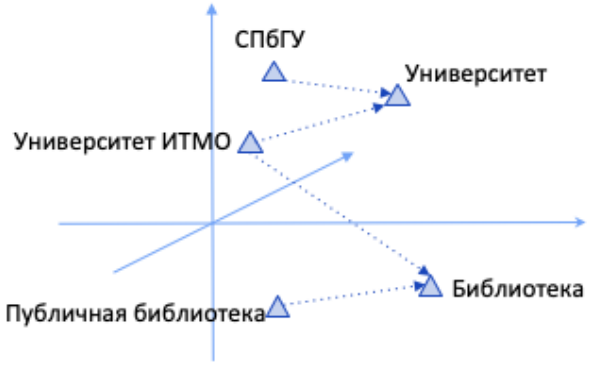
Если нам удалось закодировать узлы и связи с помощью векторов $Zu$ и $Zv$ то, тогда становится возможным вычисление семантической близости узлов в векторном пространстве как $similarity\,(u, v) \approx Zv^T Zu$




$$h + r \approx t$$

# Knowledge Graph Embeddings - KGE

Идея — минимизация функции расстояния между верными и схожими утверждениями, и максимизация между ложными.




- Разные функции расстояния
- 50-75-мерные пространства
- Семантика скрыта в векторе
- Сложность интерпретации
- Больше размер графа — точнее векторные представления
- Вычислительно затратно

# KGE - Graphs as Tensors



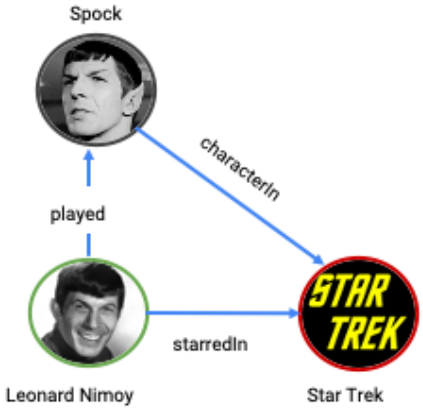

Nickel et al. A review of relational machine learning for knowledge graphs. IEEE. 2015

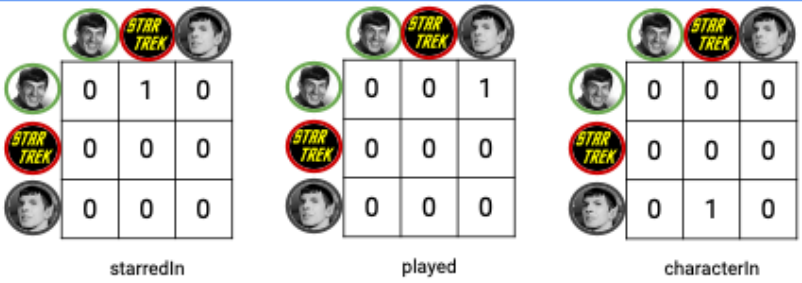
# KGE - Graphs as Tensors



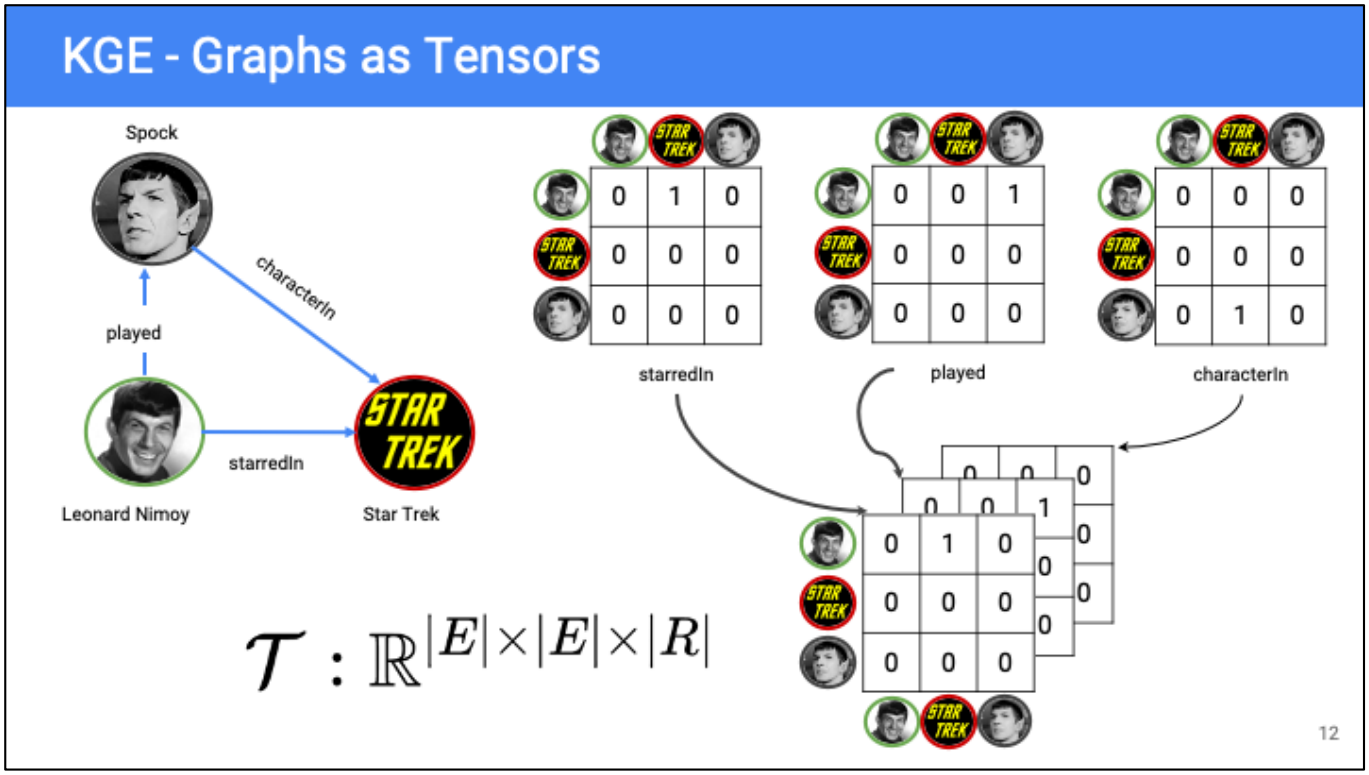
Элементарная матрица для представления связи starredIn

# KGE - Graphs as Tensors




Матрицы для графа, включающего три связи

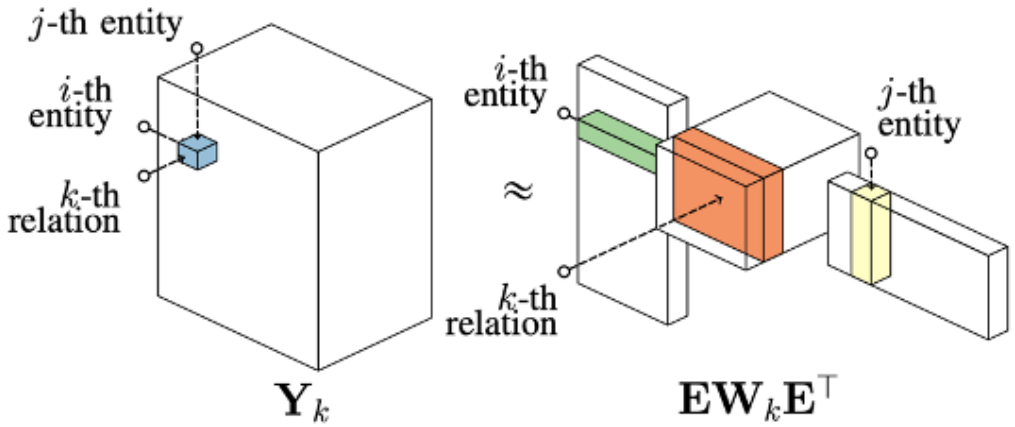



Матрицы всех отношений могут быть совмещены в общий тензор. Элемент тензора $y_{ijk} = 1$ обозначает факт, что существует отношение (*i-th entity, k-th predicate, j-th entity*). В противном случае, для несуществующих или неизвестных отношений элемент приравнивается к нулю.

## KGE – алгоритм RESCAL

**Tensor Factorization**

Задача тензорной факторизации - разложить трехмерный тензор на векторы E (узлы) и R (связи)




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

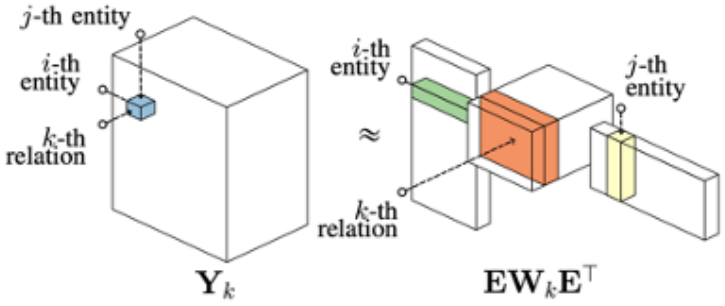
13

Для моделирования бинарных отношений на графе удобно использовать трёхсторонний тензор **Y**, в котором две моды образованы идентично на основе конкатенированных сущностей объектов-узлов, а третья мода содержит отношения между ними Подобный подход получил название тензорная факторизация.

## KGE – алгоритм RESCAL

**Tensor Factorization**

Задача тензорной факторизации - разложить трехмерный тензор на векторы E (узлы) и R (связи)




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

14

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

Сущности ГЗ могут быть эффективно представлены векторами их латентных свойств. Данные свойства называют латентными, т.к. они напрямую не описаны в данных, но могут быть выведены из имеющихся данных в процессе МО. В работе [29] предложена модель графовых латентных свойств *RESCAL*, представляющая тройки посредством парного взаимодействия этих латентных свойств. Вычисление вероятности существования какой-либо тройки в ГЗ осуществляется с помощью специальной оценочной функции. Тензорное представление графа позволяет эффективным образом вычислять подобные оценки через факторизацию срезов тензора $\boldsymbol{F_k}$. является матрицей, содержащей все оценки для *k*-й связи (отношения) и *i*-го ряда в матрице $\boldsymbol{E}$. $W^k$является матрицей весов, элементы которой $w_{abk}$ показывают, насколько латентные свойства $a$ и $b$ взаимосвязаны в *k*-том отношении