

Описание метода в статье Improving Knowledge Graph Embeddings with Ontological Reasoning

В статье «Improving Knowledge Graph Embeddings with Ontological Reasoning» авторы предлагают метод ReasonKGE, который использует онтологическое рассуждение для улучшения качества встраиваний графов знаний (Knowledge Graph Embeddings, KGE). Основная цель — повысить семантическую корректность предсказаний, избегая генерации несогласованных (inconsistent) троек, нарушающих аксиомы онтологии.

Модели встраивания графов знаний (например, TransE, ComplEx) обучаются на положительных тройках (фактах) и искусственно сгенерированных отрицательных тройках (предположительно ложных).

Проблема методики: Отрицательные примеры часто генерируются случайно или по упрощённым правилам (например, замена субъекта/объекта). Это может привести к тому, что модель учится на ложных отрицательных примерах (на самом деле верных фактах), или, наоборот, предсказывает семантически невозможные тройки, нарушающие онтологию.

Решение: использовать онтологию для:

1. Выявления несогласованных предсказаний, сделанных моделью.
2. Превращения таких предсказаний в отрицательные примеры для дообучения модели.
3. Обобщения этих отрицательных примеров на семантически похожие случаи.

Это делается итеративно, в цикле «предсказать → проверить согласованность → добавить отрицательные примеры → переобучить».

Ключевые компоненты метода ReasonKGE

Онтология в DL-LiteSt

Используется легковесная онтология на основе DL-Lite с расширениями (транзитивность и дизъюнкция классов):

- Позволяет эффективно выполнять проверку согласованности (consistency checking)
- Поддерживают конструкции: домены/области отношений, дизъюнкцию классов, транзитивность

Пример аксиом из статьи:

- $\text{Domain}(\text{worksAt}) \sqsubseteq \text{person}$ — субъект отношения `worksAt` должен быть человеком.
- $\text{Range}(\text{locatedIn}) \sqsubseteq \text{location}$ — объект `locatedIn` должен быть локацией.
- $\text{person} \sqcap \text{location} \sqsubseteq \perp$ — классы `person` и `location` несовместимы (дизъюнктны).

Итеративный процесс обучения

1. Инициализация

- Используется стандартная модель встраивания (например, TransE).
- На первой итерации отрицательные примеры генерируются обычным способом (например, случайной заменой).

2. Предсказание

- Модель предсказывает топ- k субъектов/объектов для каждого отношения в обучающем наборе.
- Получаются новые тройки, которых нет в исходном графе.

3. Проверка согласованности (Consistency Checking)

- Для каждой предсказанной тройки $\alpha = \langle s, p, o \rangle$:
- Строится подмножество графа:
 $\text{Relv}(\alpha, G) = \{\alpha\} \cup \{\beta \in G \mid \text{Ent}(\beta) \cap \text{Ent}(\alpha) \neq \emptyset\}$ (т.е. только факты, связанные с s или o).
- Проверяется, согласовано ли $\text{Relv}(\alpha, G) \cup O$ с онтологией O .
- Если несогласовано \rightarrow тройка α — некорректный предикт, и её нужно использовать как отрицательный пример.

Благодаря DL-Lite, проверка согласованности всего графа сводится к проверке небольших модулей.

4. Объяснение несогласованности

- Для каждой несогласованной тройки вычисляется минимальное объяснение (explanation):
 $E = E_G \cup E_O$, где:
 - $E_G \sqsubseteq G$ — часть графа, участвующая в конфликте.
 - $E_O \sqsubseteq O$ — аксиомы онтологии, вызывающие противоречие.

Пример:

- Предсказана тройка $\langle \text{bosch}, \text{locatedIn}, \text{john} \rangle$.
- В графе: $\langle \text{john}, \text{type}, \text{person} \rangle$.
- В онтологии: $\exists \text{locatedIn}^- \sqsubseteq \text{location}, \text{person} \sqcap \text{location} \sqsubseteq \perp$.
- Следовательно, john должен быть одновременно person и $\text{location} \rightarrow$ противоречие.

5. Обобщение отрицательных примеров (Generalization)

- Идея: если $\langle \text{bosch}, \text{locatedIn}, \text{john} \rangle$ неверно, то, вероятно, и $\langle \text{bosch}, \text{locatedIn}, \text{bob} \rangle$ тоже неверно, если bob имеет тот же тип (person).
- Для этого вводится понятие локального типа сущности: $\tau(e) = \langle \text{incoming_relations}, \text{types}, \text{outgoing_relations} \rangle$
- Обобщённые отрицательные примеры: $\text{GeneralizedSamples}(\alpha) = \{\langle s, p, o' \rangle | \tau(o, E_G \setminus \{\alpha\}) \sqsubseteq \tau(o', G)\}$
То есть: заменяем объект на любой другой, у которого локальный тип "шире или равен" типу оригинального объекта в контексте объяснения.

Это позволяет масштабировать отрицательные примеры без полного перебора.

6. Переобучение модели

- Все обобщённые несогласованные тройки добавляются как отрицательные примеры в функцию потерь.
- Модель переобучается.
- Процесс повторяется несколько итераций.

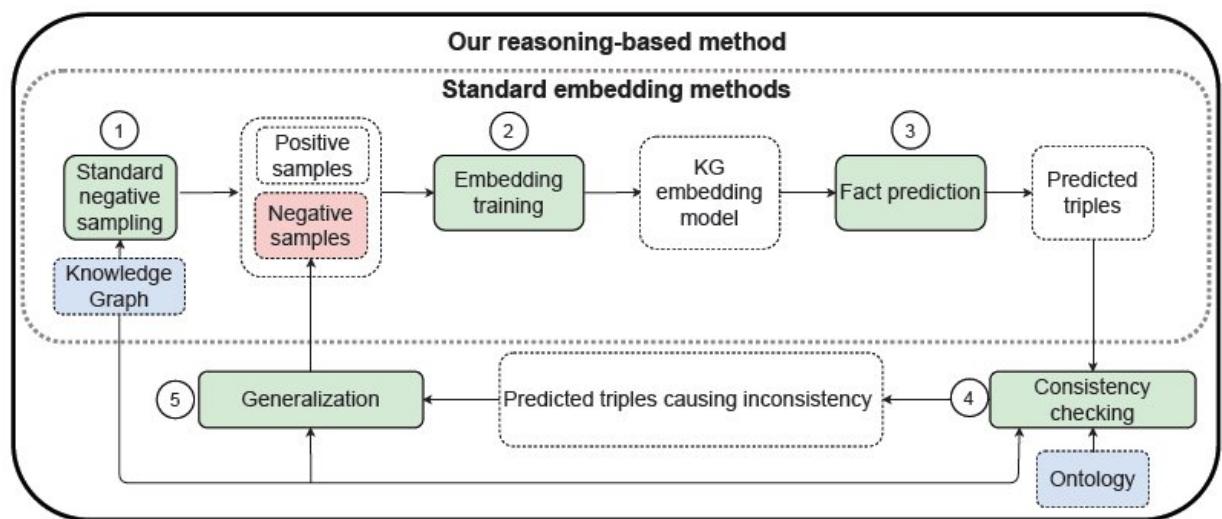
Схема работы ReasonKGE

KG + Ontology

Train baseline KGE (e.g., TransE) with random negatives

Iteratively:

1. Predict top-k triples
2. For each prediction:
 - a. Build relevant subgraph
 - b. Check consistency with ontology
 - c. If inconsistent → get explanation
 - d. Generalize to similar negative samples
3. Retrain KGE with new negatives



ReasonKGE — это гибридный нейросимволический подход, где:

- Нейросеть (KGE) генерирует гипотезы.
- Reasoner проверяет их на логическую корректность.
- Обратная связь от reasoner используется для улучшения обучения нейросети.

Преимущества подхода

ReasonKGE позволяет:

1. Динамически, генерировать отрицательные примеры используя реальные ошибки модели
2. Сохраняет масштабируемость благодаря локальности DL-Lite
3. Предсказания согласованы с онтологией, меньше бессмысленных фактов

Эксперименты показывают:

- Улучшение MRR, Hits@k по сравнению с базовыми моделями.
- Снижение доли несогласованных предсказаний (Inc@k) — иногда в десятки раз.

Table 3: Link prediction results

Model	KG	Default Training			Static Sampling			ReasonKGE		
		MRR Hits@1 Hits@10		MRR Hits@1 Hits@10		MRR Hits@1 Hits@10		MRR Hits@1 Hits@10		MRR Hits@1 Hits@10
TransE	LUBM3U	0.119	0.069	0.214	0.125	0.082	0.213	0.135	0.079	0.256
	Yago3-10	0.226	0.044	0.537	0.351	0.183	0.621	0.367	0.197	0.629
	DBpedia15k	0.109	0.061	0.206	0.101	0.073	0.254	0.118	0.101	0.299
ComplEx	LUBM3U	0.159	0.119	0.242	0.181	0.136	0.276	0.233	0.195	0.313
	Yago3-10	0.482	0.400	0.643	0.515	0.431	0.665	0.530	0.453	0.668
	DBpedia15k	0.099	0.061	0.174	0.098	0.107	0.193	0.115	0.125	0.221

Table 4: Ratio of inconsistent predictions (the lower, the better).

Model	KG	Prediction	Default Training		Static Sampling		ReasonKGE	
			Inc@1	Inc@10	Inc@1	Inc@10	Inc@1	Inc@10
TransE	LUBM3U	subject	0.169	0.270	0.428	0.250	0.037	0.133
		object	0.095	0.097	0.212	0.104	0.005	0.007
ComplEx	YAGO3-10	subject	0.075	0.280	0.629	0.492	0.075	0.273
		object	0.026	0.136	0.114	0.089	0.020	0.117
	DBpedia15K	subject	0.311	0.652	0.401	0.663	0.217	0.585
		object	0.413	0.538	0.428	0.544	0.170	0.460
	LUBM3U	subject	0.041	0.097	0.177	0.136	0.036	0.069
		object	0.008	0.012	0.003	0.007	0.005	0.007
	YAGO3-10	subject	0.113	0.198	0.169	0.128	0.071	0.143
		object	0.037	0.115	0.065	0.084	0.015	0.074
	DBpedia15K	subject	0.488	0.667	0.436	0.695	0.344	0.583
		object	0.397	0.585	0.365	0.528	0.318	0.533

Активация \

Это позволяет совместить масштабируемость машинного обучения с гарантированной семантической корректностью, что особенно важно в приложениях, где точность и логическая согласованность критичны (медицина, финансы, интеллектуальные ассистенты и др.).