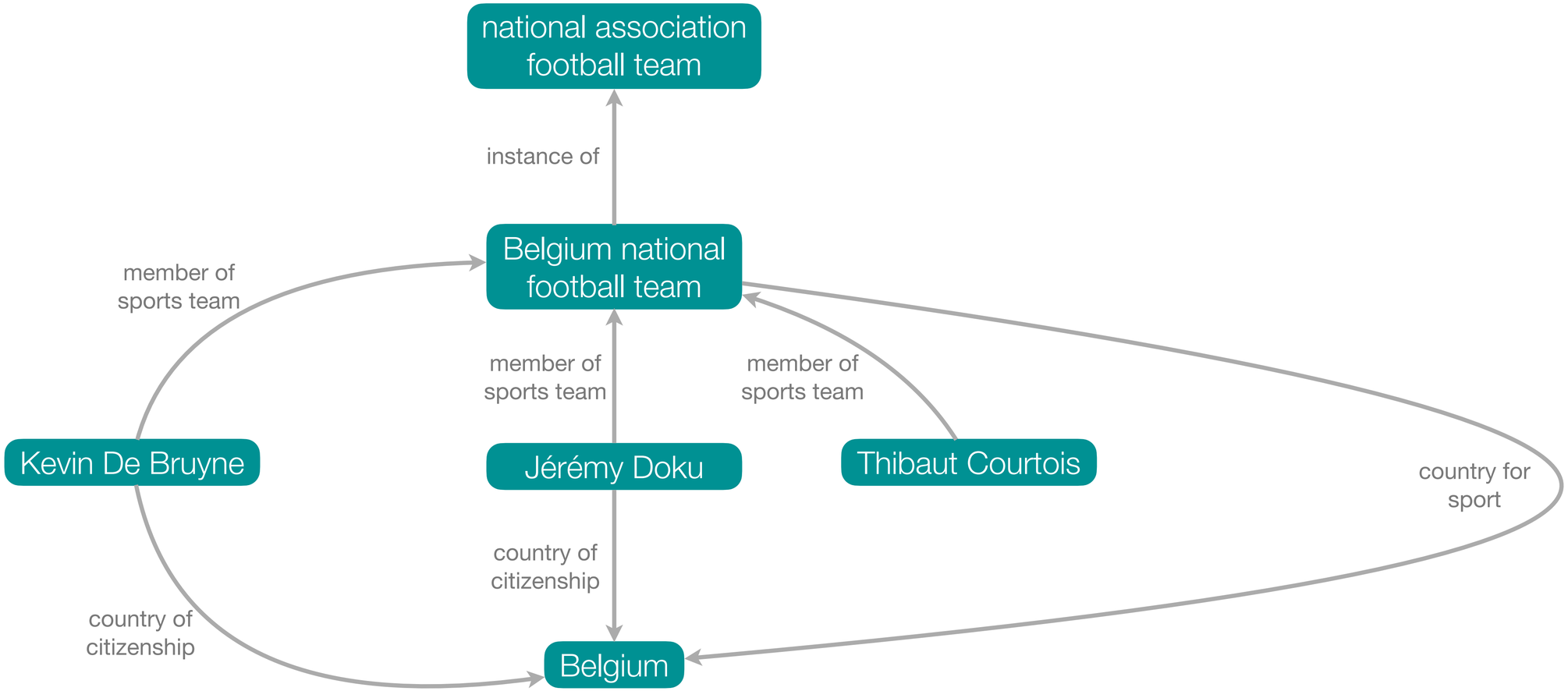
Geometric Representations of Relational Knowledge for Neuro-symbolic Reasoning

Steven Schockaert

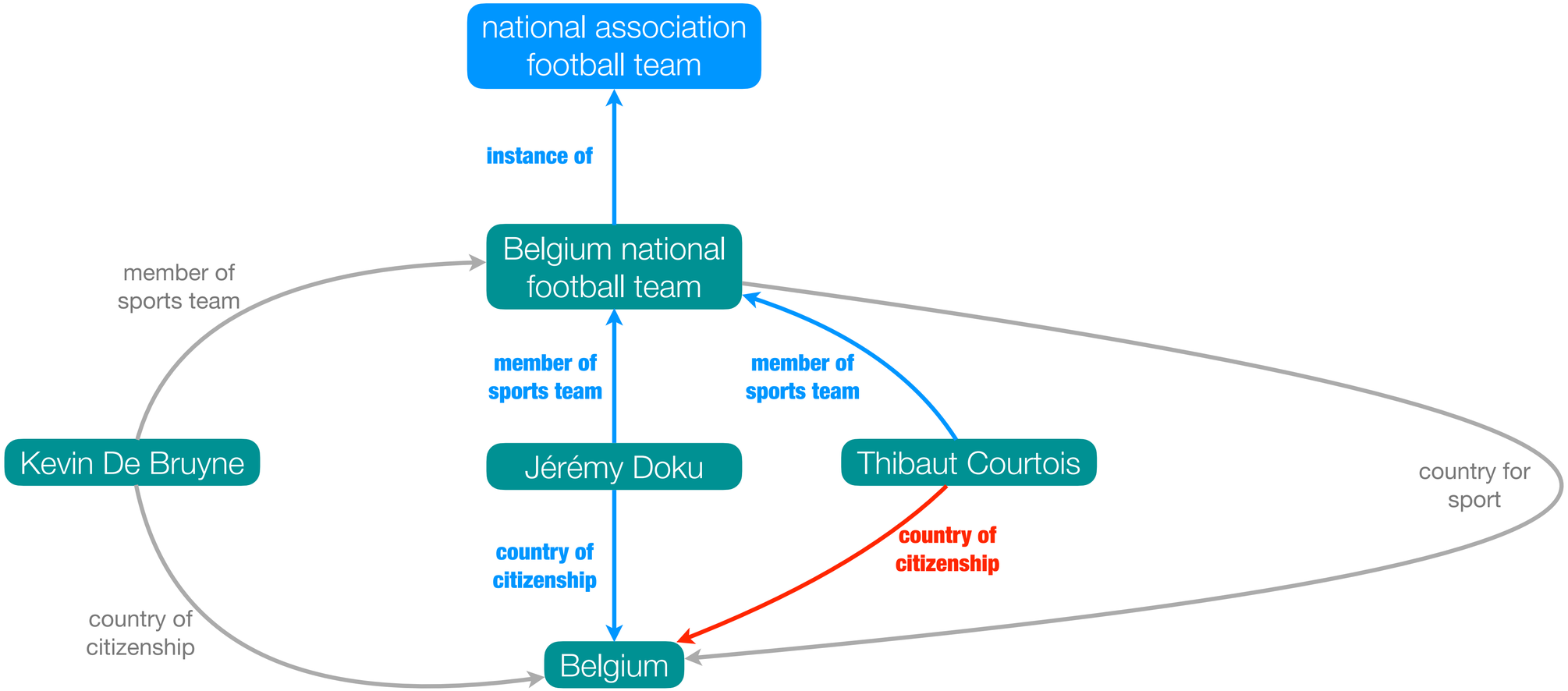
Cardiff University, UK
schockaerts1@cardiff.ac.uk

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Reasoning in knowledge graphs



Reasoning in knowledge graphs



Rule-based approaches

Train: learn rules that capture the regularities that are observed in the (training) graph

Test: select the rule with the highest confidence degree that can answer the query

$$\text{hasNationality}(X_2, Z) \leftarrow \text{memberOfTeam}(X_1, Y) \wedge \text{memberOfTeam}(X_2, Y) \wedge \text{instanceOf}(Y, \text{nationalFootballTeam}) \wedge \text{hasNationality}(X_1, Z)$$

Rule-based approaches

Train: learn rules that capture the regularities that are observed in the (training) graph

Test: select the rule with the highest confidence degree that can answer the query

$$\text{hasNationality}(X_2, Z) \leftarrow \text{memberOfTeam}(X_1, Y) \wedge \text{memberOfTeam}(X_2, Y) \wedge \text{instanceOf}(Y, \text{nationalFootballTeam}) \wedge \text{hasNationality}(X_1, Z)$$

Advantages: efficient and transparent

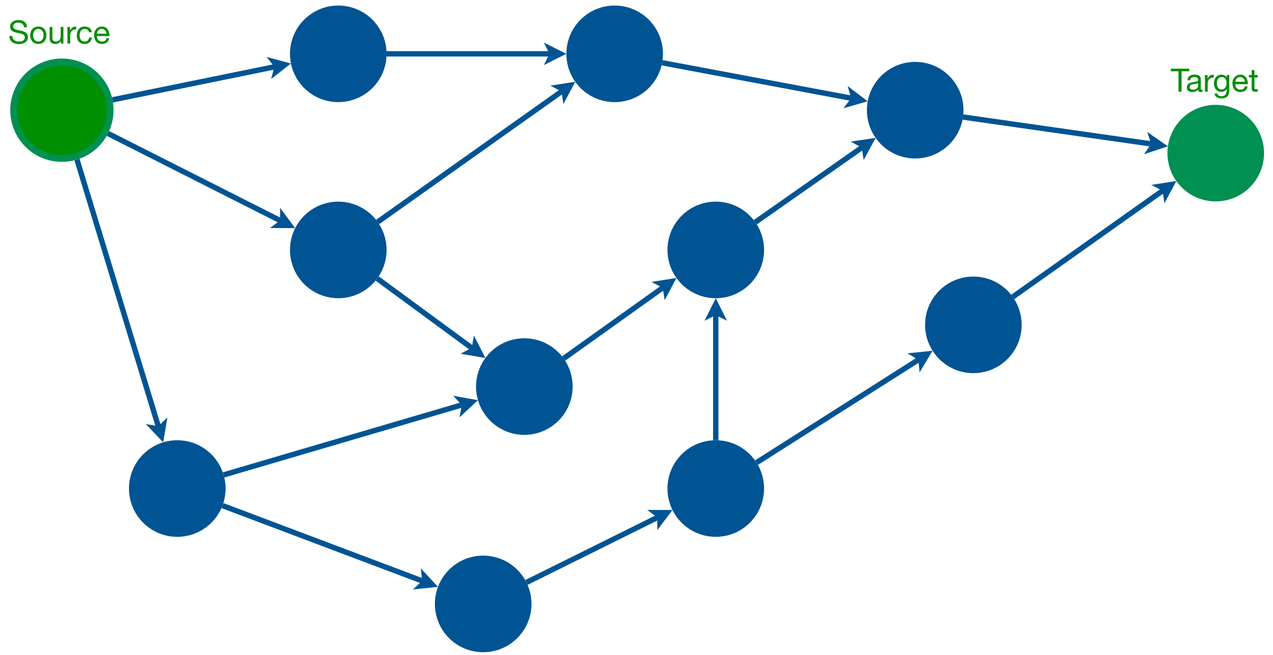
Disadvantages:

- Rule learning is hard (e.g. predicate invention)
- No generalisation beyond rule bodies that are (frequently) observed in the training data
- Learned rules often too “brittle” for systematic reasoning (usually only one-off rule application)
- Difficult to aggregate multiple pieces of evidence (when multiple ground rules can be applied)
- Difficult to integrate with neural models

Graph neural networks

Graph neural networks

Fix source and target entities and treat link prediction as a graph classification problem

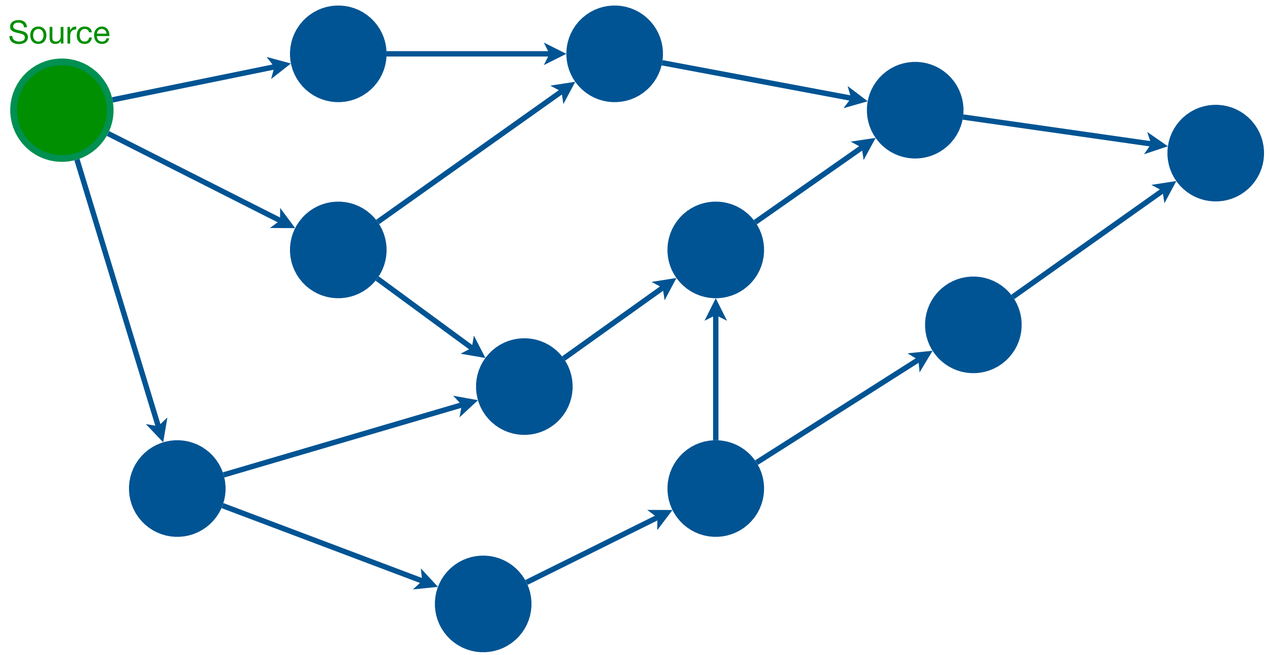


Disadvantage: inefficient (GNN inference for each candidate answer of every query)

Komal K. Teru, Etienne G. Denis, William L. Hamilton: Inductive Relation Prediction by Subgraph Reasoning. ICML 2020: 9448-9457

Graph neural networks

Fix source entity and learn representations that capture the relationship to that entity

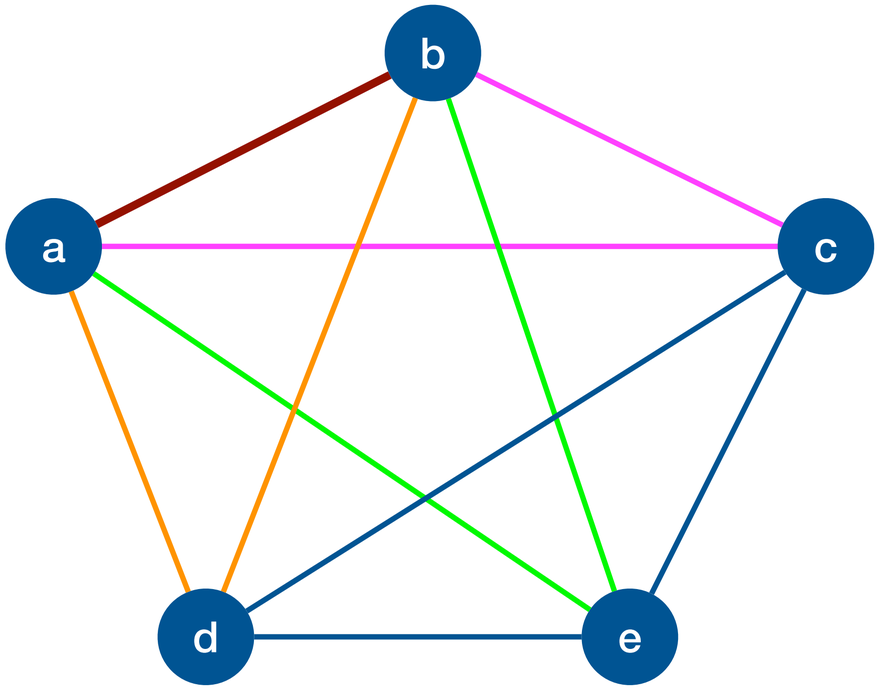


Disadvantage: we still need to repeat GNN inference for every query

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, Jian Tang: Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS 2021: 29476-29490

Edge transformers

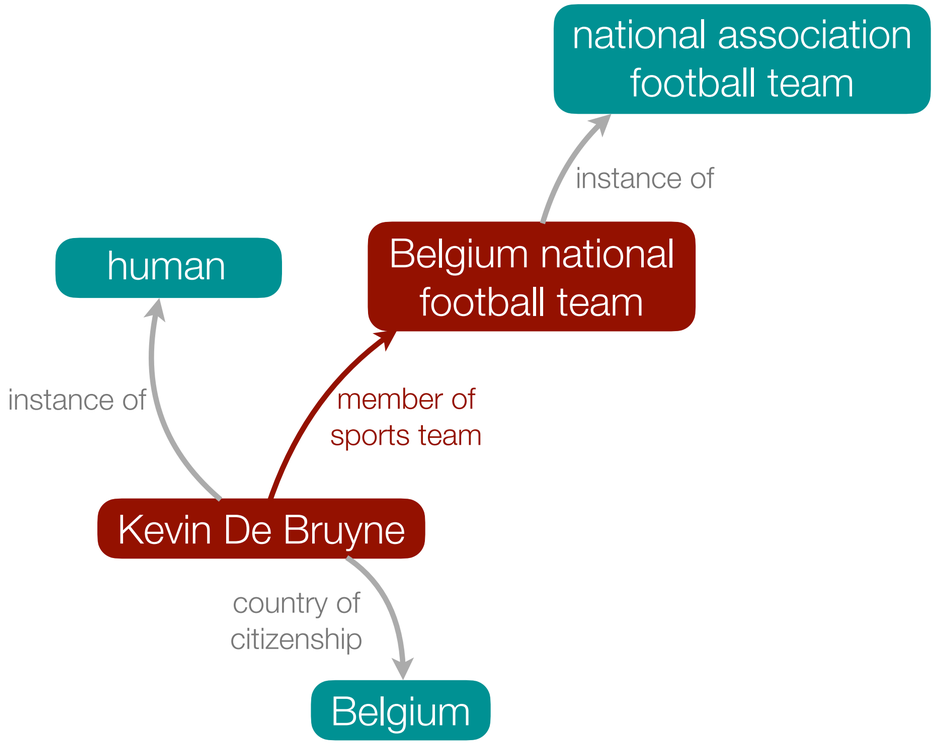
Learn a vector for every pair of entities, update vectors by computing all “relational compositions”

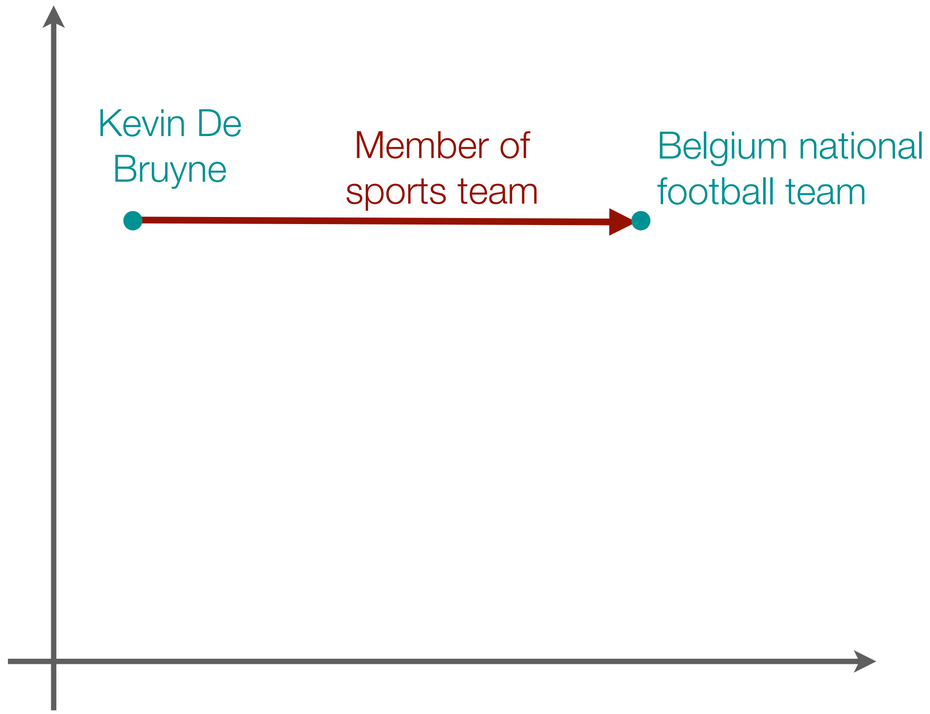


Disadvantage: need to consider every triple of entities in each update step

Leon Bergen, Timothy J. O'Donnell, Dzmitry Bahdanau: Systematic Generalization with Edge Transformers. NeurIPS 2021: 1390-1402

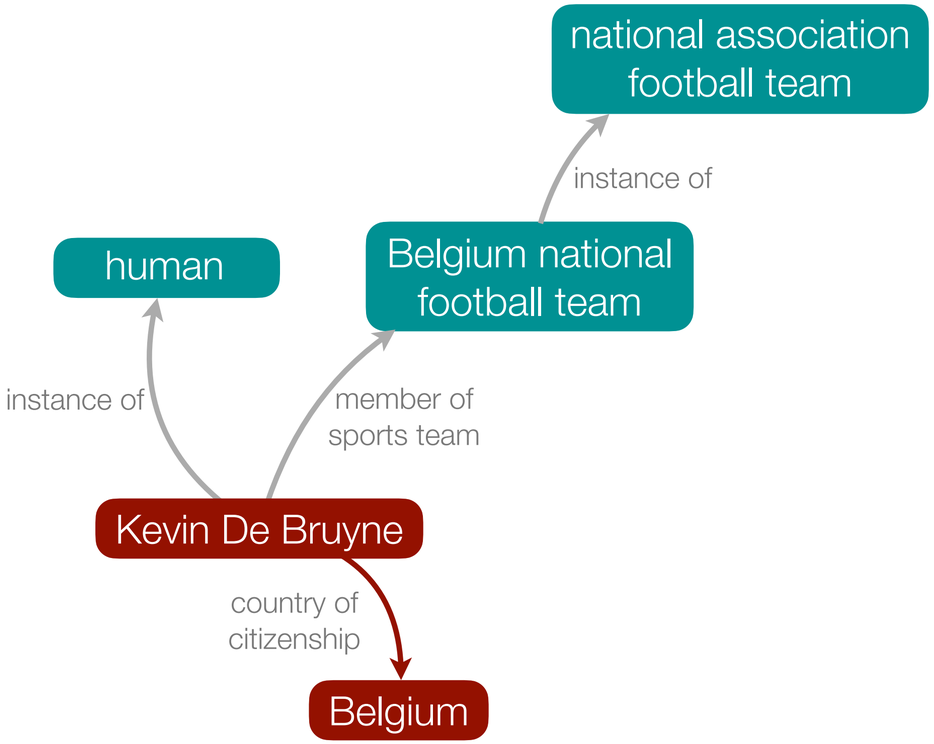
Knowledge graph embeddings

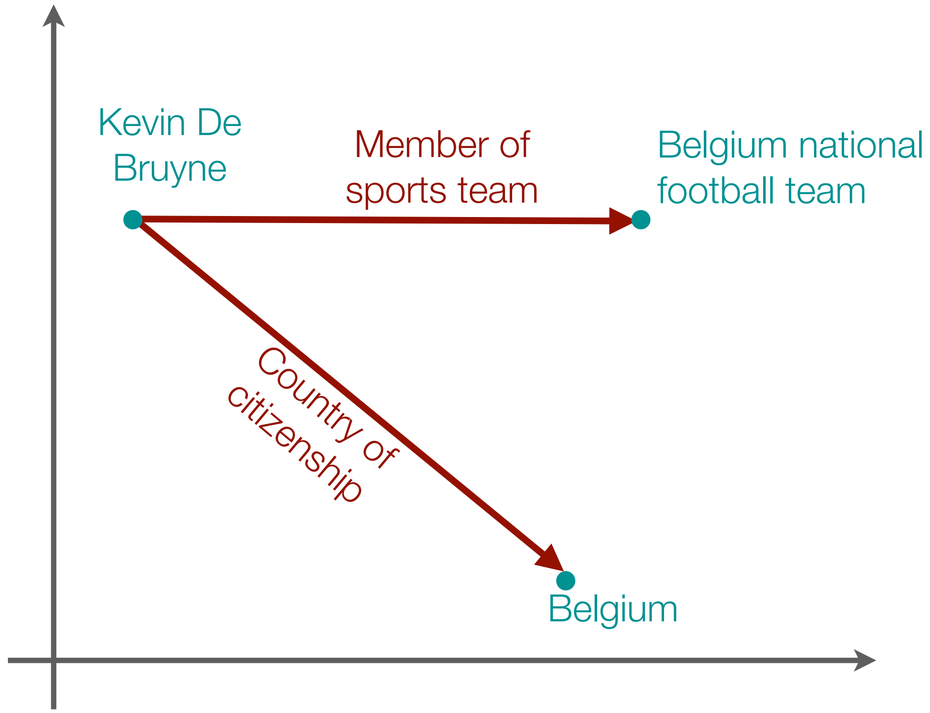




Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, Oksana Yakhnenko: Translating Embeddings for Modeling Multi-relational Data. NIPS 2013: 2787-2795

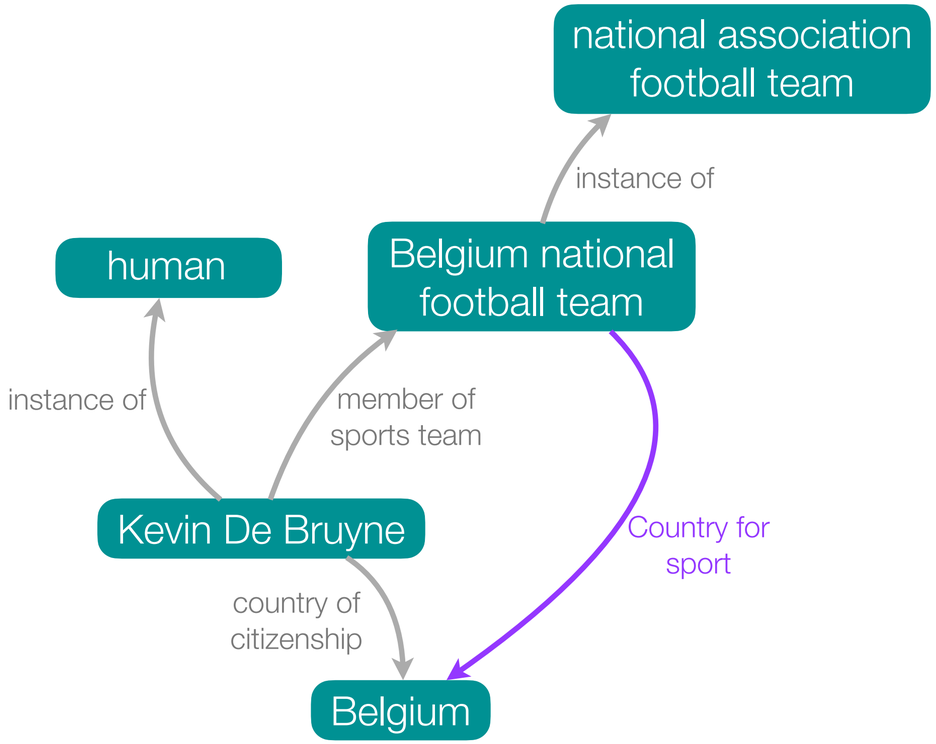
Knowledge graph embeddings

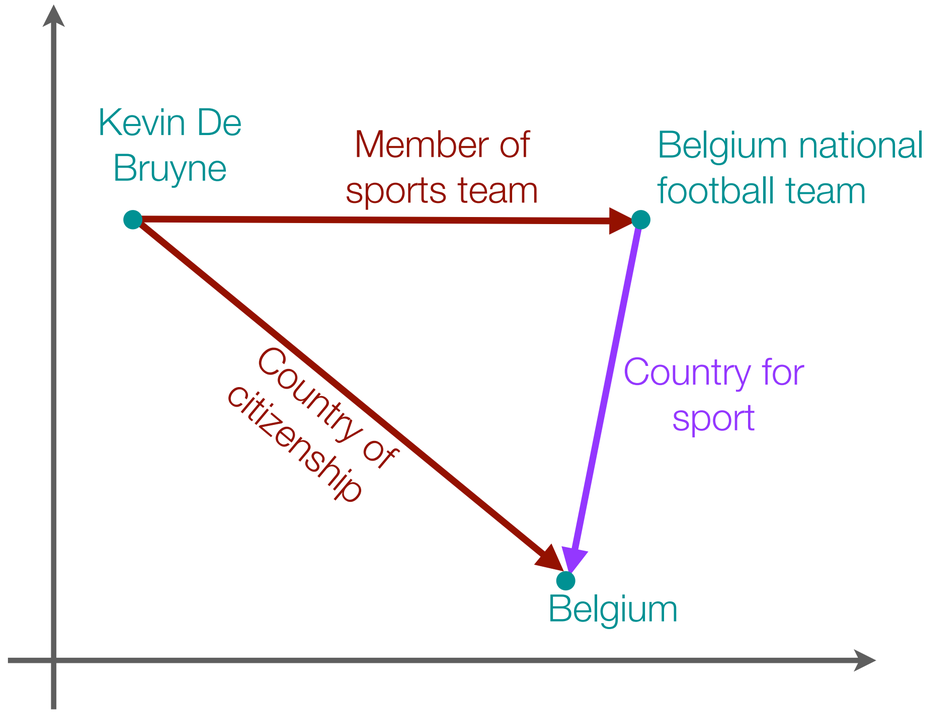




Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, Oksana Yakhnenko: Translating Embeddings for Modeling Multi-relational Data. NIPS 2013: 2787-2795

Knowledge graph embeddings





Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, Oksana Yakhnenko: Translating Embeddings for Modeling Multi-relational Data. NIPS 2013: 2787-2795

Knowledge graph embeddings

Note: Embeddings could be learned directly or produced by a GNN (e.g. for inductive KG completion)

Advantages:

- Highly scalable
- Vector representations are convenient for retrieval and for interacting with other neural models

Disadvantages:

- Unclear what kinds of reasoning different models are capable of

Knowledge graph embeddings

Note: Embeddings could be learned directly or produced by a GNN (e.g. for inductive KG completion)

Advantages:

- Highly scalable
- Vector representations are convenient for retrieval and for interacting with other neural models

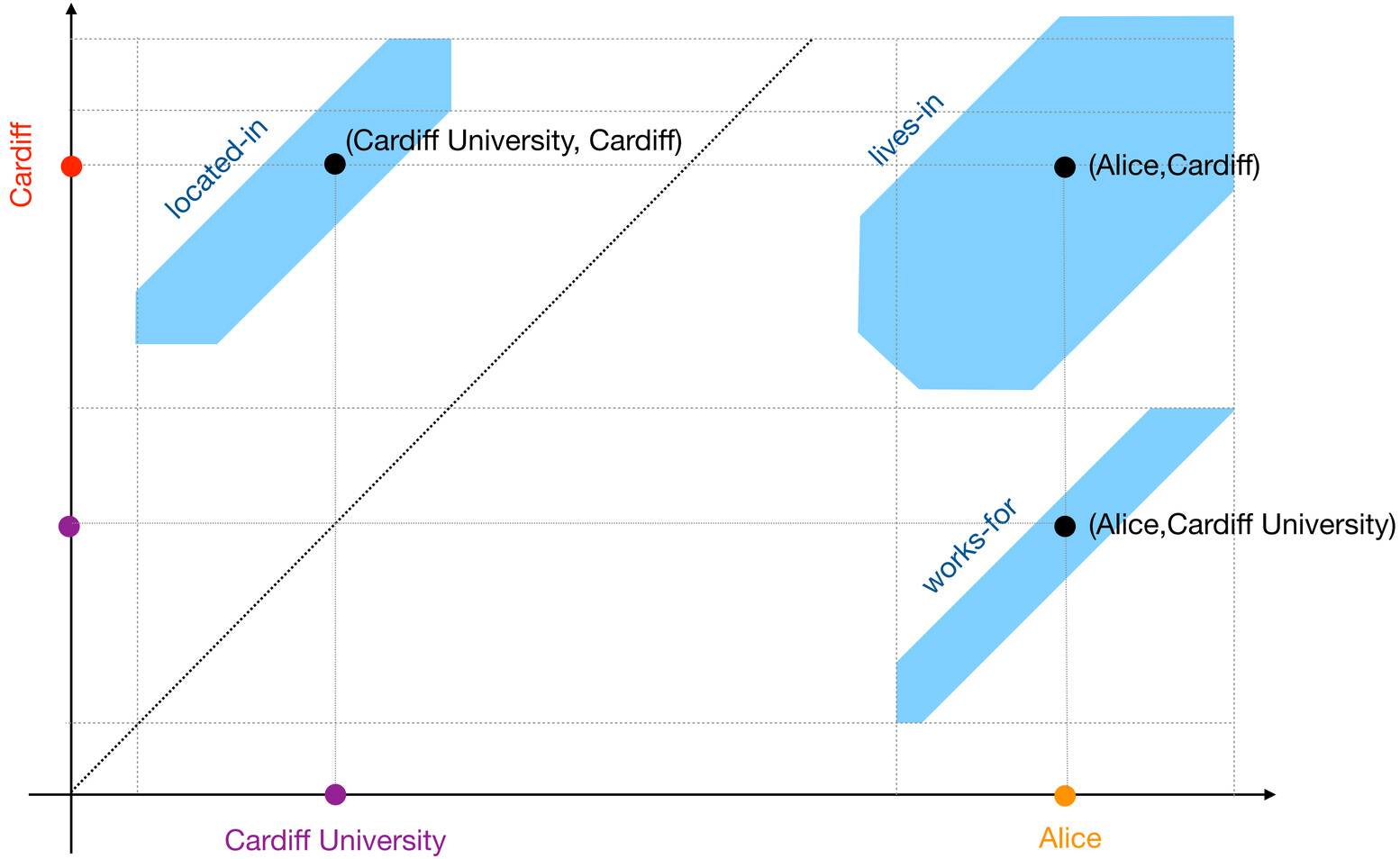
Disadvantages:

- Unclear what kinds of reasoning different models are capable of

Focus of this talk

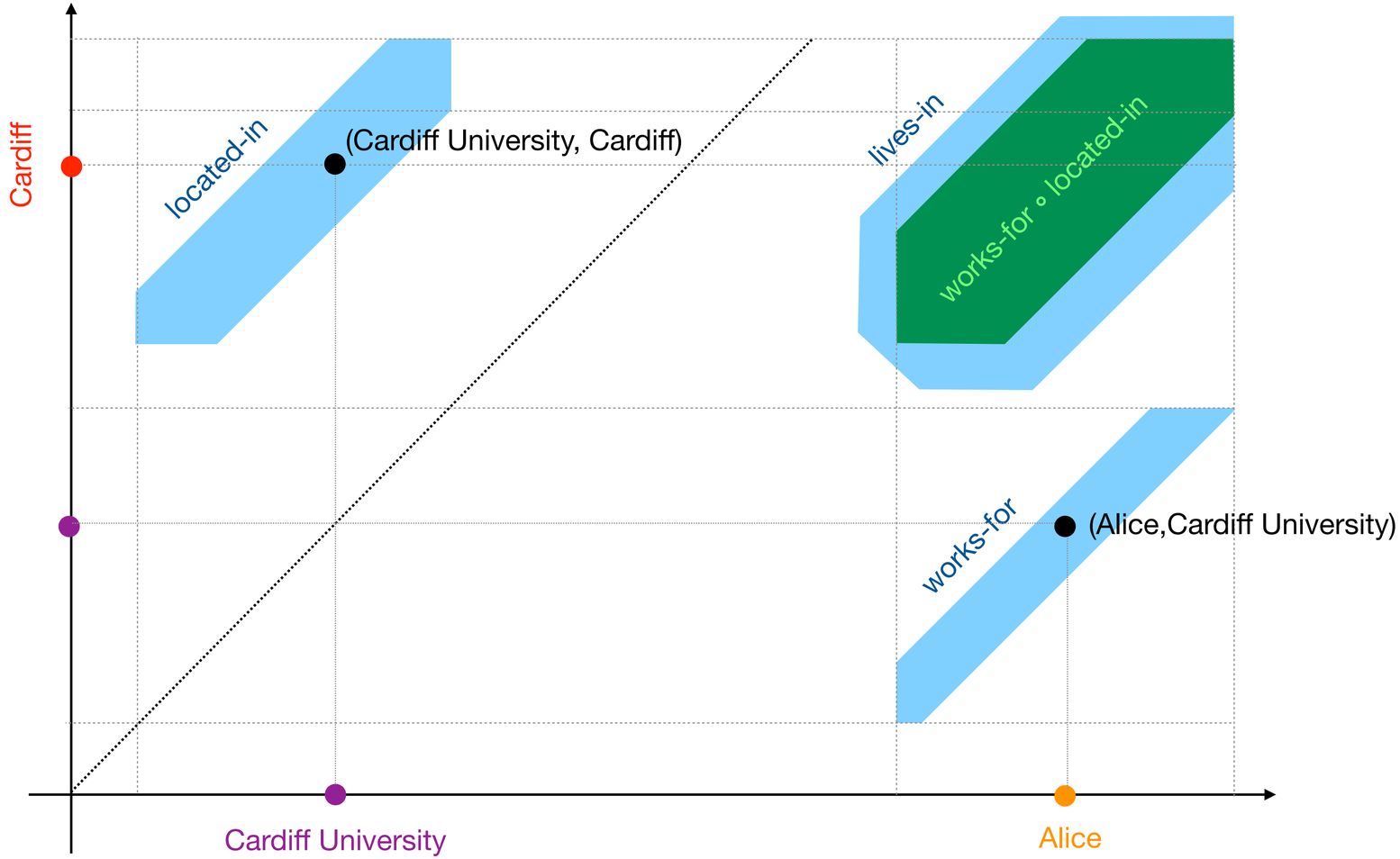
Region based embeddings

Region-based relation embeddings



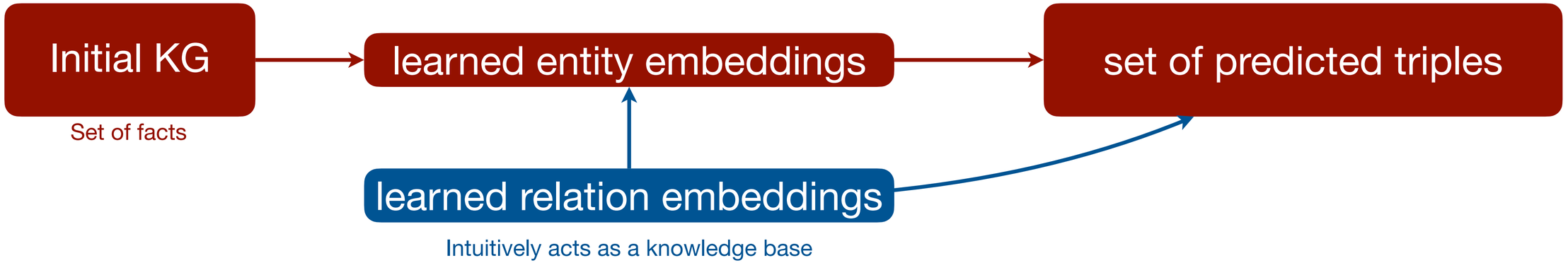
Víctor Gutiérrez-Basulto, Steven Schockaert: From Knowledge Graph Embedding to Ontology Embedding? An Analysis of the Compatibility between Vector Space Representations and Rules. KR 2018

Region-based relation embeddings

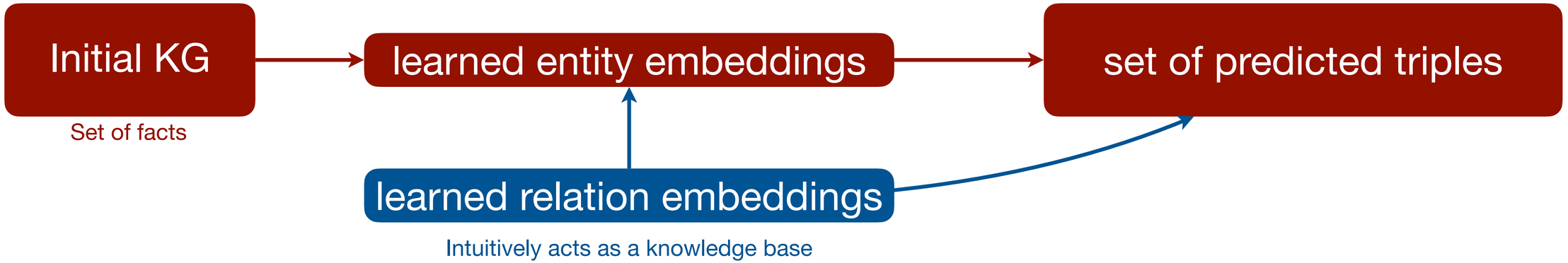


$$\textit{works-for}(X, Y) \wedge \textit{located-in}(Y, Z) \rightarrow \textit{lives-in}(X, Z)$$

Reasoning with relation embeddings

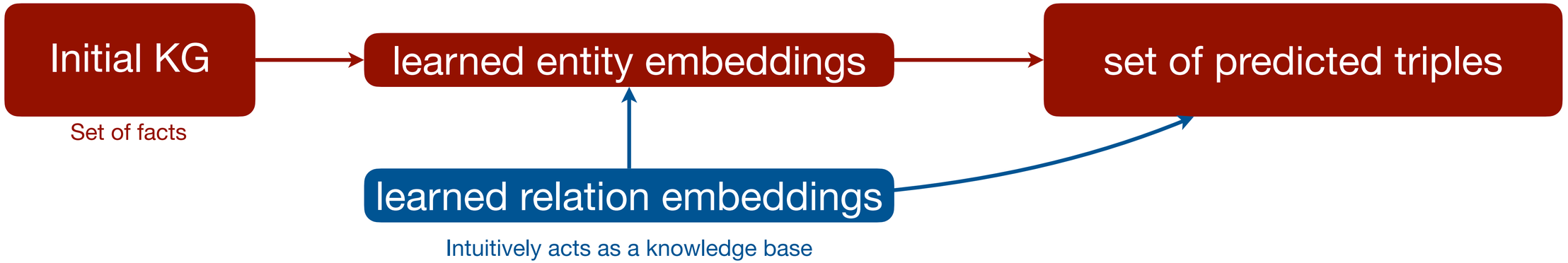


Reasoning with relation embeddings



Central question: do there exist relation embeddings for which this inference process is equivalent to reasoning with a given rule base?

Reasoning with relation embeddings



Central question: do there exist relation embeddings for which this inference process is equivalent to reasoning with a given rule base?

Given a rule base K, do there exist relation embeddings such that **for every knowledge graph G and relational fact $r(a, b)$**, we have:

$$G \cup K \models r(a, b) \quad \longleftrightarrow$$

Any entity embeddings capturing the relational facts from G (w.r.t. the given relation embeddings) also capture $r(a, b)$

Reasoning with relation embeddings

In practice we typically don’t have any rule base. Our main focus is on understanding what kinds of rule bases embedding models can learn in principle (but if some rules are known a priori we still want to be able to “inject” them into the learning process)

Reasoning with relation embeddings

In practice we typically don’t have any rule base. Our main focus is on understanding what kinds of rule bases embedding models can learn in principle (but if some rules are known a priori we still want to be able to “inject” them into the learning process)

Usually clear how a given rule can be captured by relation embeddings (i.e. how we can guarantee that the consequences of a rule are captured whenever its premises are captured). The challenge is to ensure that **only** those rules entailed by the knowledge base are captured

Reasoning with relation embeddings

In practice we typically don't have any rule base. Our main focus is on understanding what kinds of rule bases embedding models can learn in principle (but if some rules are known a priori we still want to be able to "inject" them into the learning process)

Usually clear how a given rule can be captured by relation embeddings (i.e. how we can guarantee that the consequences of a rule are captured whenever its premises are captured). The challenge is to ensure that **only** those rules entailed by the knowledge base are captured

We essentially want to find the **simplest** class of relation embeddings that is capable of capturing a given type of knowledge

- Simple models like TransE often outperform more sophisticated ones in practice
- We ideally want models that can be easily integrated into neuro-symbolic systems

Coordinate-wise regions

$$\mathbf{e} = (e_1, \ldots, e_n)$$

$$\mathbf{f} = (f_1, \ldots, f_n)$$

$$\eta(r) = \{(e_1, \ldots, e_n, f_1, \ldots, f_n) \mid \}$$

Region-based representation of r (2n-dimensional)

Coordinate-wise regions

Two-dimensional region $\eta_n(r)$

$$\mathbf{e} = (e_1, \ldots, e_n)$$
$$\mathbf{f} = (f_1, \ldots, f_n)$$

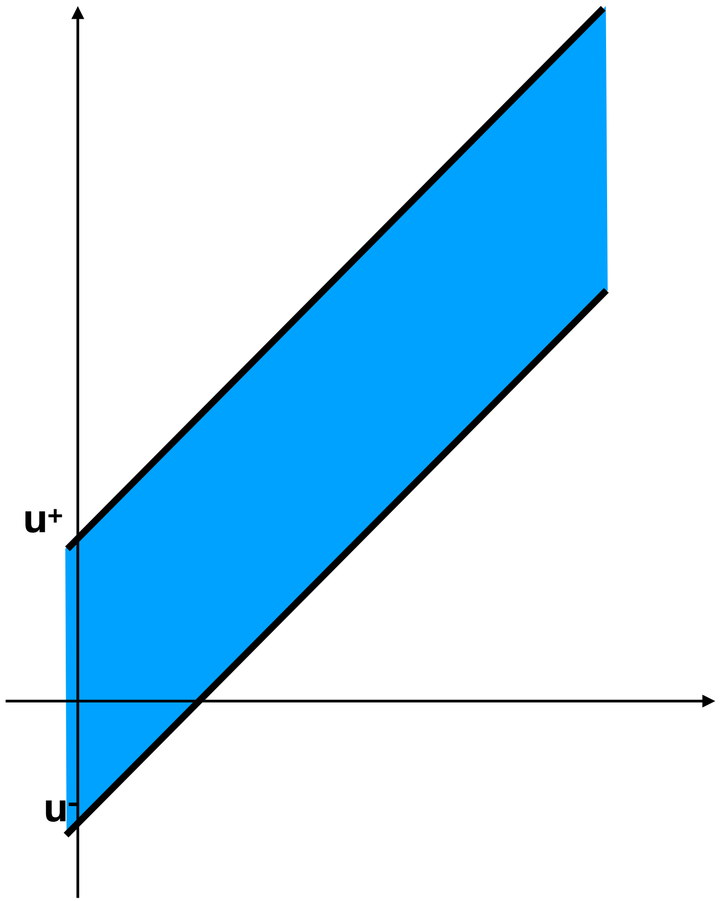
Two-dimensional region $\eta_1(r)$

$$\eta(r) = \{(e_1, \ldots, e_n, f_1, \ldots, f_n) \mid (e_1, f_1) \in \eta_1(r), \ldots, (e_n, f_n) \in \eta_n(r)\}$$

Region-based representation of r (2n-dimensional)

Region used to compare the first coordinates of the entity vectors (2-dimensional)

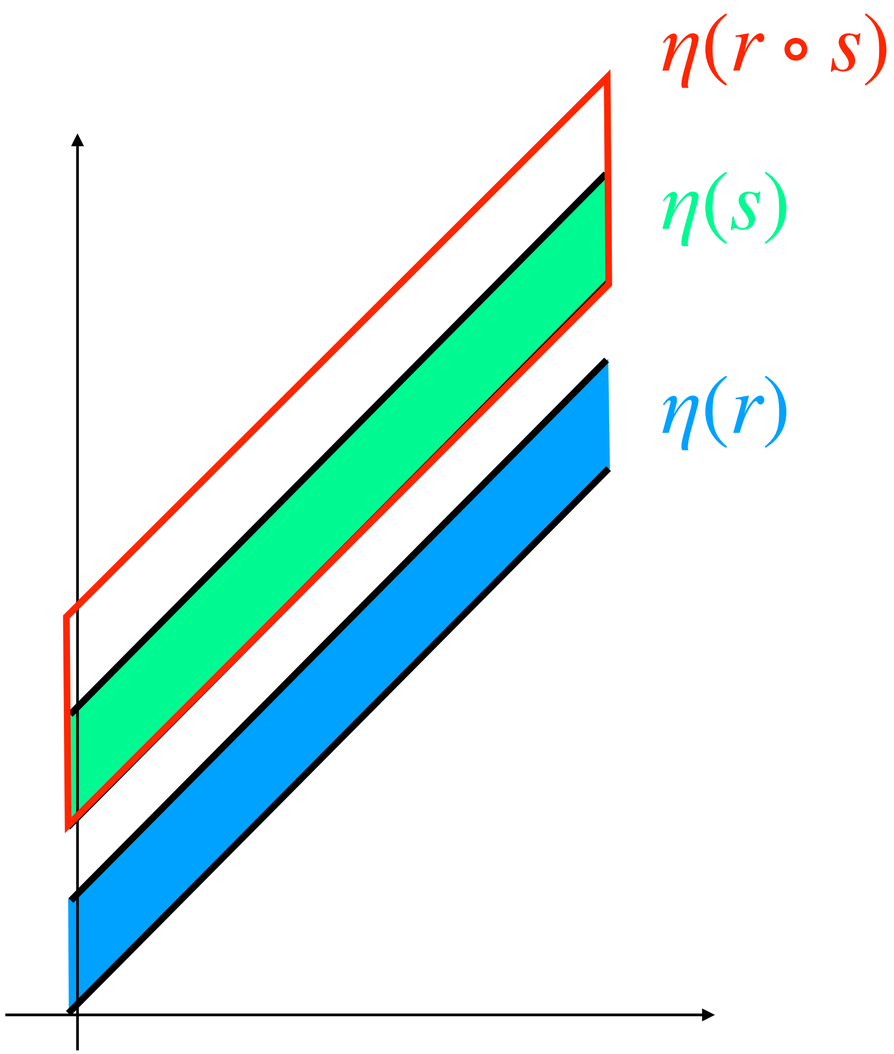
TransE regions



Regions are defined using two parameters per coordinate (a lower bound u^- and an upper bound u^+)

Compared to TransE, the width of the regions allows us to capture relation hierarchies

TransE regions

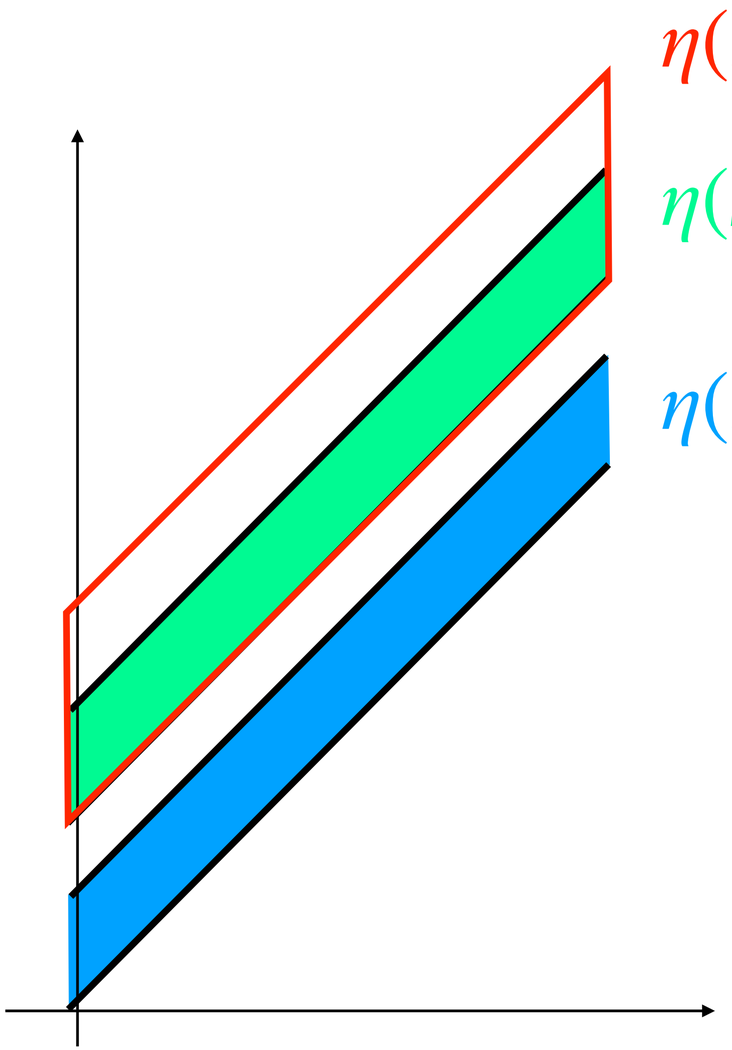


Compositions are easy to compute:

Lower bound of $\eta(r \circ s)$ is the sum of the lower bounds of $\eta(r)$ and $\eta(s)$

Upper bound of $\eta(r \circ s)$ is the sum of the upper bounds of $\eta(r)$ and $\eta(s)$

TransE regions



But: this implies $\eta(r \circ s) = \eta(s \circ r)$

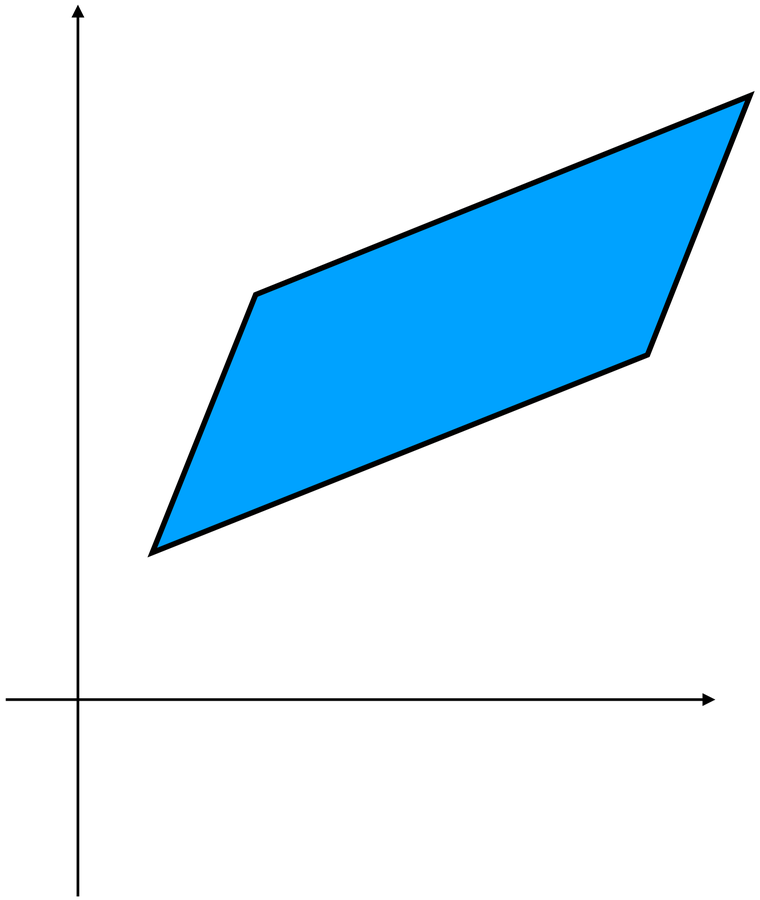
This means that whenever we capture a rule like:

$$\textit{mother}(X, Y) \wedge \textit{brother}(Y, Z) \rightarrow \textit{uncle}(X, Z)$$

We would also capture the (semantically invalid) symmetric counterpart:

$$\textit{brother}(X, Y) \wedge \textit{mother}(Y, Z) \rightarrow \textit{uncle}(X, Z)$$

ExpressivE

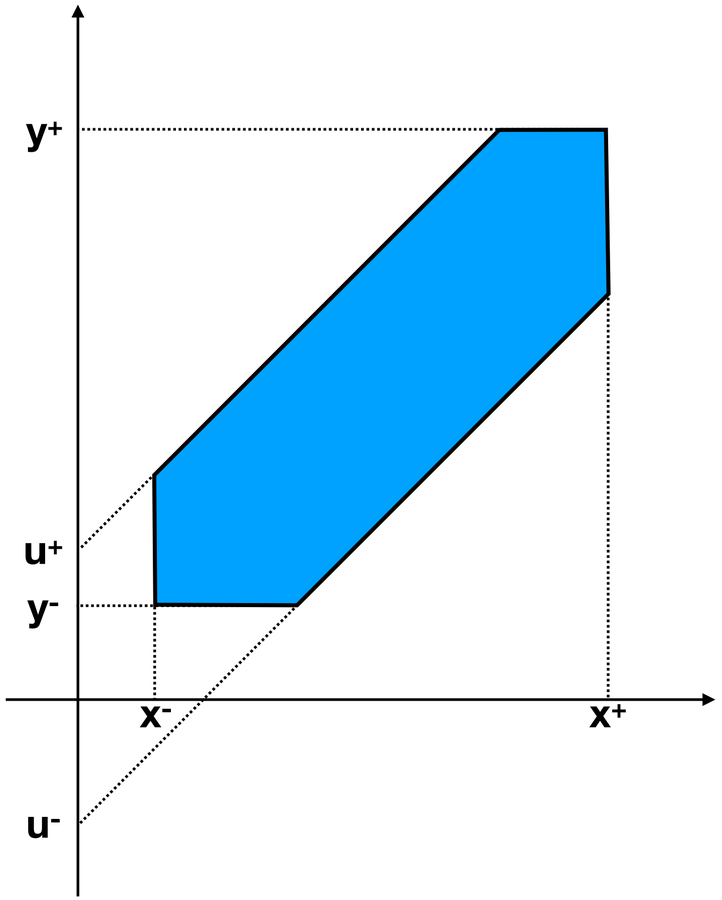


Parallelograms avoid the symmetry of composition

But: parallelograms are not closed under composition and intersection

Aleksandar Pavlovic, Emanuel Sallinger: ExpressivE: A Spatio-Functional Embedding For Knowledge Graph Completion. ICLR 2023

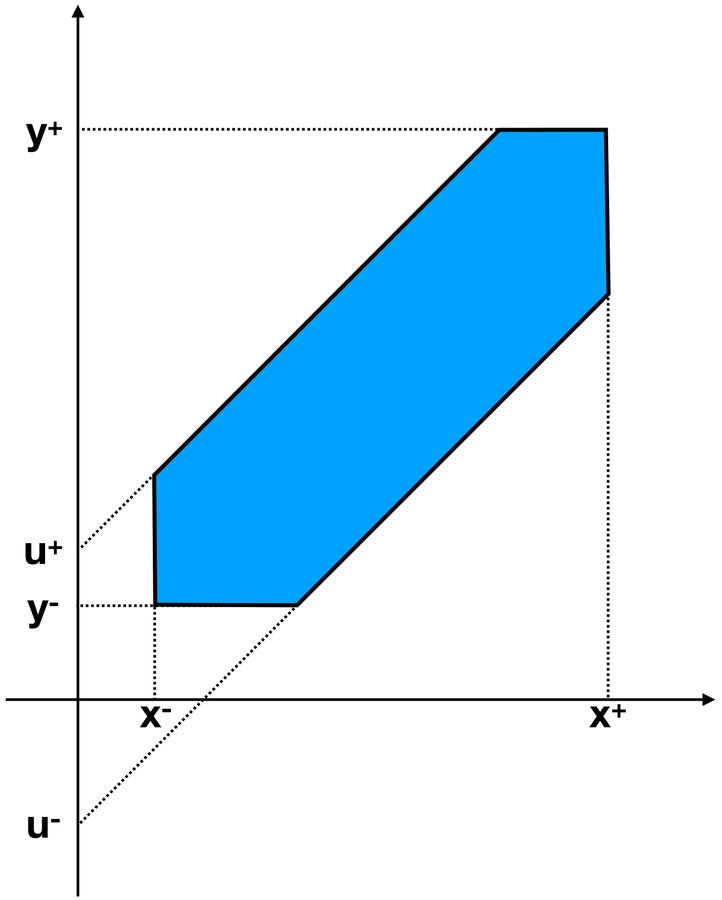
Modelling relations with hexagons



Can we make TransE more expressive by adding domain and range constraints?

Victor Charpenay, Steven Schockaert: Capturing Knowledge Graphs and Rules with Octagon Embeddings. IJCAI 2024

Modelling relations with hexagons



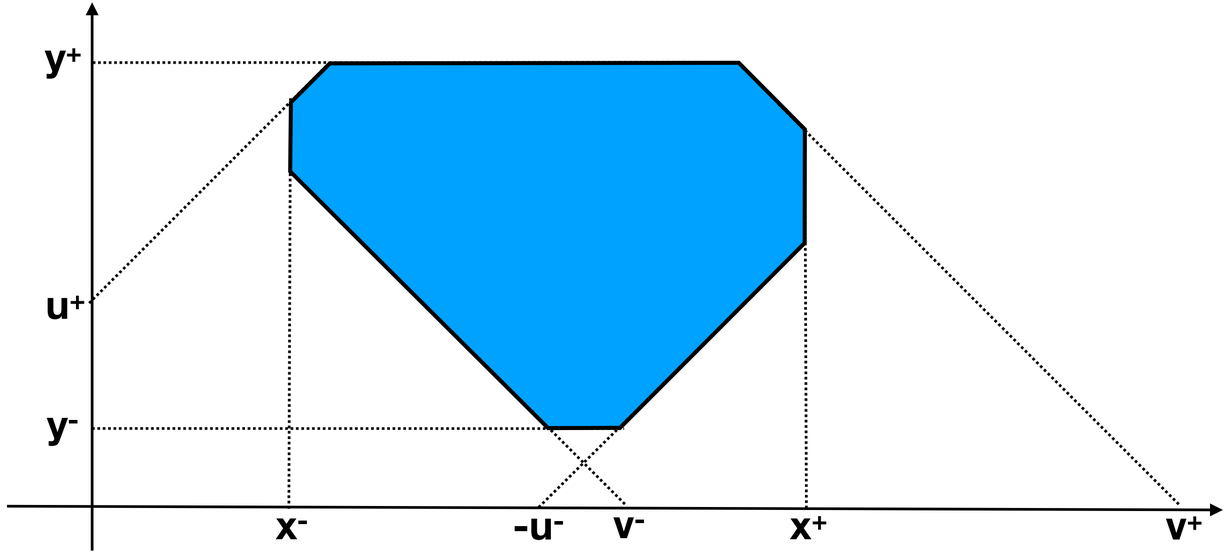
Can we make TransE more expressive by adding domain and range constraints?

- Closed under intersection and composition
- Not possible to capture arbitrary knowledge graphs

If (e,r,f) and (f,r,e) are captured then so is (e,r,e)

Victor Charpenay, Steven Schockaert: Capturing Knowledge Graphs and Rules with Octagon Embeddings. IJCAI 2024

Modelling relations with octagons



Adding further constraints we obtain axis-aligned octagons

- Closed under intersection and composition
- Possible to capture arbitrary knowledge graphs

Victor Charpenay, Steven Schockaert: Capturing Knowledge Graphs and Rules with Octagon Embeddings. IJCAI 2024

Capturing rules with octagons

Focus on closed path rules:

$$r_1(X_1, X_2) \wedge \ldots \wedge r_k(X_k, X_{k+1}) \rightarrow r_{k+1}(X_1, X_{k+1})$$

We call such a rule **regular** if $r_i \neq r_j$ for $i \neq j$

$$\mathit{mother}(X, Y) \wedge \mathit{brother}(Y, Z) \rightarrow \mathit{uncle}(X, Z)$$

regular

$$\mathit{father}(X, Y) \wedge \mathit{father}(Y, Z) \rightarrow \mathit{grandfather}(X, Z)$$

$$\mathit{brother}(X, Y) \wedge \mathit{sister}(Y, Z) \rightarrow \mathit{sister}(X, Z)$$

not regular

Victor Charpenay, Steven Schockaert: Capturing Knowledge Graphs and Rules with Octagon Embeddings. IJCAI 2024

Capturing rules with octagons

Focus on closed path rules:

$$r_1(X_1, X_2) \wedge \ldots \wedge r_k(X_k, X_{k+1}) \rightarrow r_{k+1}(X_1, X_{k+1})$$

We call such a rule **regular** if $r_i \neq r_j$ for $i \neq j$

Proposition: Let K be a set of **regular rules** such that every rule entailed from K is either a regular rule or a trivial rule. There exists an octagon embedding which **captures exactly** those rules that are entailed by K.

Can we also model non-regular rules?

Suppose an embedding model with **convex coordinate-wise regions** captures the following rules:

$$r_1 \subseteq r_2 \qquad r_1 \circ r_1 \circ r_1 \subseteq r_2$$
$$r_1 \circ r_1 \subseteq r_3 \qquad r_1 \circ r_1 \circ r_1 \subseteq r_3$$
$$r_1 \subseteq r_4 \qquad r_2 \circ r_2 \subseteq r_4$$
$$r_3 \circ r_3 \subseteq r_4$$

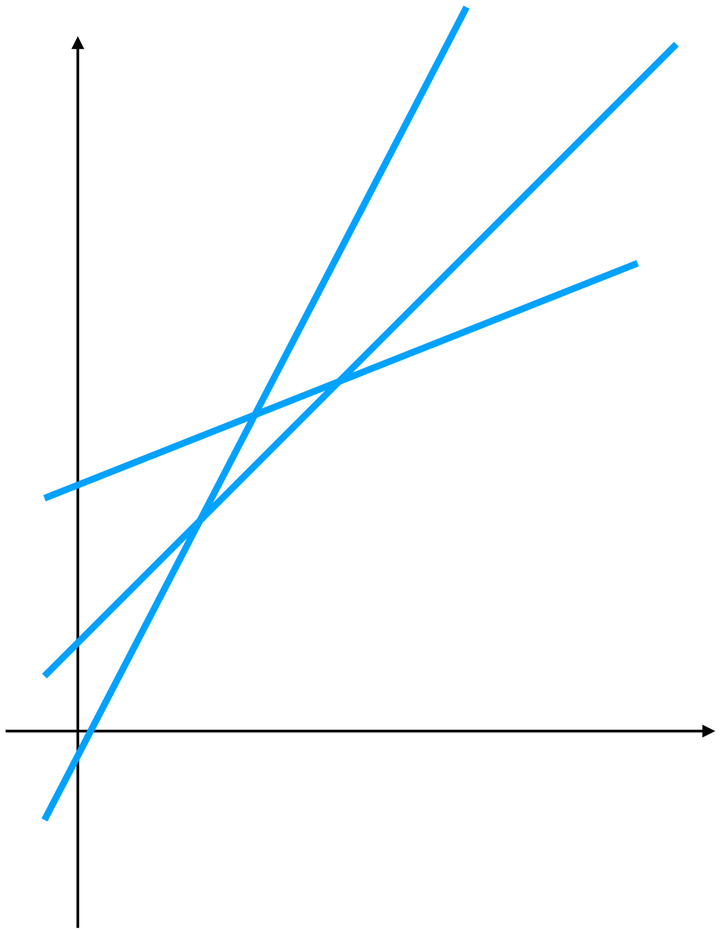
Then this model also captures the following rule:

$$r_1 \circ r_1 \circ r_1 \subseteq r_4$$

Victor Charpenay, Steven Schockaert: Faithful Relational Reasoning with Region-based Embeddings: Expressivity of Convex Coordinate-wise Models. ICML 2026

Escaping the limitations of convex coordinate-wise models

Non-convex coordinate-wise models



Consider a coordinate-wise model where regions are modelled, in each coordinate, as a union of lines

For any set of closed path rules P, there exists such a coordinate-wise model that captures exactly the set of rules that can be inferred in m steps from P

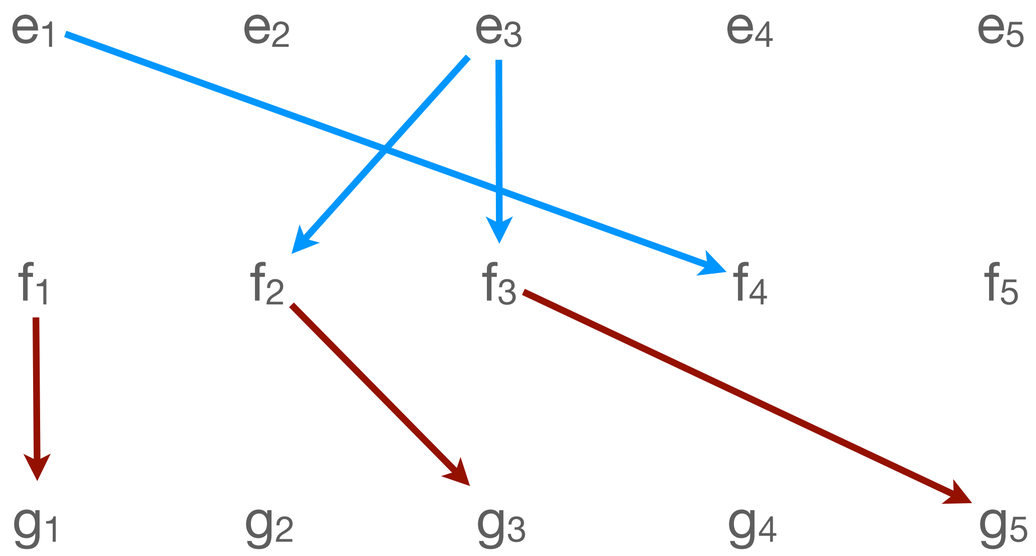
Going beyond coordinate-wise regions

Set of coordinate indices

$$\mathbf{e} \oplus \mathbf{f} \in X_r \quad \text{iff} \quad \forall i \in I_r . e_{\sigma_r(i)} \leq f_i$$

Mapping from coordinate indices to coordinate indices

Going beyond coordinate-wise regions

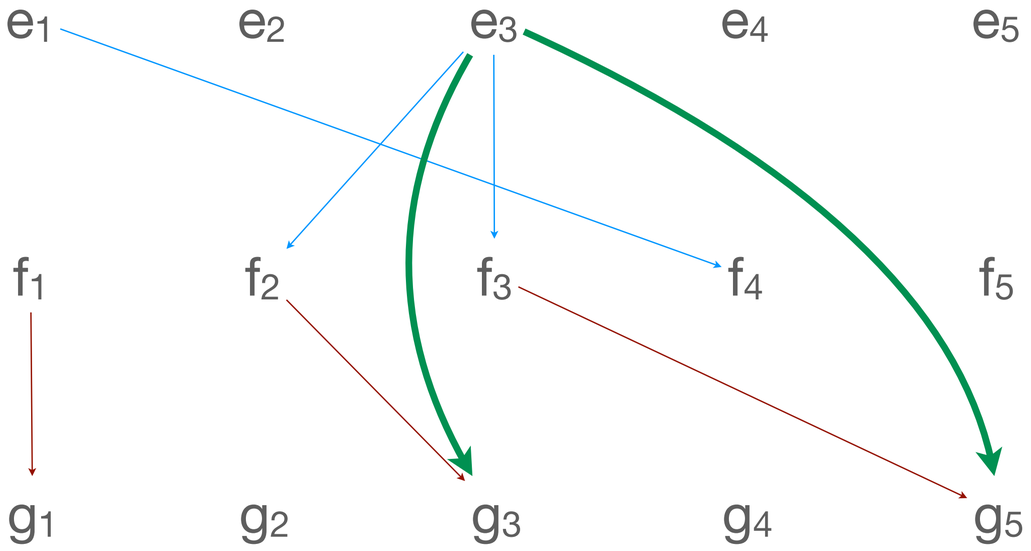


Ordering constraints modelling relation r

Ordering constraints modelling relation s

Aleksandar Pavlovic, Emanuel Sallinger, Steven Schockaert: Faithful Differentiable Reasoning with Reshuffled Region-based Embeddings. KR 2025

Going beyond coordinate-wise regions



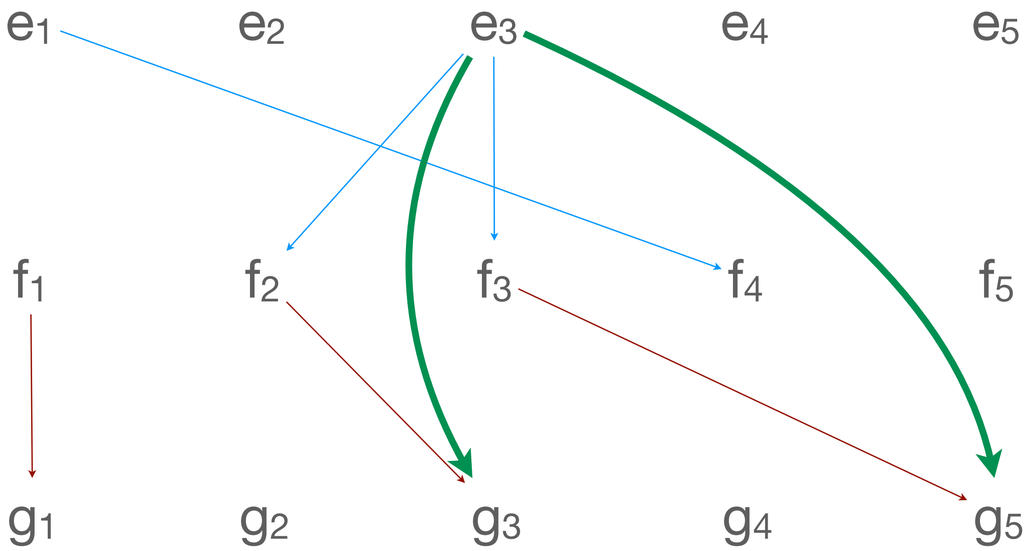
Constraints implied by $r(e, f) \wedge s(f, g)$

If relation t is modelled by a subset of these constraints, then the following rule is captured:

$$r(X, Y) \wedge s(Y, Z) \rightarrow t(X, Z)$$

Aleksandar Pavlovic, Emanuel Sallinger, Steven Schockaert: Faithful Differentiable Reasoning with Reshuffled Region-based Embeddings. KR 2025

Going beyond coordinate-wise regions



Constraints implied by $r(e, f) \wedge s(f, g)$

If relation t is modelled by a subset of these constraints, then the following rule is captured:

$$r(X, Y) \wedge s(Y, Z) \rightarrow t(X, Z)$$

For any set of closed path rules P, there exist ordering constraints that capture exactly the set of rules that can be inferred in m steps from P

Aleksandar Pavlovic, Emanuel Sallinger, Steven Schockaert: Faithful Differentiable Reasoning with Reshuffled Region-based Embeddings. KR 2025

RESHUFFLE: learning embeddings with GNNs

Let $\mathbf{e} = (e_1, \ldots, e_n)$ and $\mathbf{f} = (f_1, \ldots, f_n)$. Then we have:

$$\forall i \in I_r \,.\, e_{\sigma_r(i)} \leq f_i \quad \text{iff} \quad \max(\mathbf{A_r e}, \mathbf{f}) = \mathbf{f}$$

where all entries in the matrix $\mathbf{A_r}$ are 0 or 1, and there is at most one non-zero component on every row.

Aleksandar Pavlovic, Emanuel Sallinger, Steven Schockaert: Faithful Differentiable Reasoning with Reshuffled Region-based Embeddings. KR 2025

RESHUFFLE: learning embeddings with GNNs

Let $\mathbf{e} = (e_1, \ldots, e_n)$ and $\mathbf{f} = (f_1, \ldots, f_n)$. Then we have:

$$\forall i \in I_r \,.\, e_{\sigma_r(i)} \leq f_i \quad \text{iff} \quad \max(\mathbf{A_r e}, \mathbf{f}) = \mathbf{f}$$

where all entries in the matrix $\mathbf{A_r}$ are 0 or 1, and there is at most one non-zero component on every row.

We can learn embeddings using a GNN of the following form

$$\mathbf{f^{(l+1)}} = \max\left(\{\mathbf{f^{(l)}}\} \cup \{\mathbf{A_r e^{(l)}} \mid (e, r, f) \in \mathcal{G}\}\right)$$

where the initial embeddings are initialised randomly.

Aleksandar Pavlovic, Emanuel Sallinger, Steven Schockaert: Faithful Differentiable Reasoning with Reshuffled Region-based Embeddings. KR 2025

RESHUFFLE: learning embeddings with GNNs

We can capture arbitrary sets of closed path rules if we limit the number of iterations of the GNN.

Proposition: Let K be an arbitrary set of closed-path rules and G a knowledge graph. There exists a parameterisation of the GNN and an $m \in \mathbb{N}$ such that after m iterations of the GNN, we obtain embeddings satisfying:

- Every triple that can be inferred from K and G in m steps is captured by the resulting embeddings
- For any triple that cannot be inferred from K and G in m steps, the probability that it is captured can be made arbitrarily small (by choosing the number of dimensions to be sufficiently high)

Aleksandar Pavlovic, Emanuel Sallinger, Steven Schockaert: Faithful Differentiable Reasoning with Reshuffled Region-based Embeddings. KR 2025

Modelling rules with encoder-decoder models

Encoder: enforcing that all triples from the KG are satisfied w.r.t. some region-based relation embedding η

Decoder: verifying which other triples are satisfied w.r.t. some region-based relation embedding η'

Victor Charpenay, Steven Schockaert: Faithful Relational Reasoning with Region-based Embeddings: Expressivity of Convex Coordinate-wise Models. ICML 2026

Modelling rules with encoder-decoder models

Encoder: enforcing that all triples from the KG are satisfied w.r.t. some region-based relation embedding η

GNN encoder: message-passing updates can sometimes be interpreted as gradient-based optimisation steps of **region-based relation embeddings model**

Decoder: verifying which other triples are satisfied w.r.t. some region-based relation embedding η'

Region-based scoring functions can be used predict triples from the learned entity embeddings

Victor Charpenay, Steven Schockaert: Faithful Relational Reasoning with Region-based Embeddings: Expressivity of Convex Coordinate-wise Models. ICML 2026

Modelling rules with encoder-decoder models

Encoder: enforcing that all triples from the KG are satisfied w.r.t. some region-based relation embedding η

Decoder: verifying which other triples are satisfied w.r.t. some region-based relation embedding η'

GNN encoder: message-passing updates can sometimes be interpreted as gradient-based optimisation steps of **region-based relation embeddings model**

These region-based embedding models don't need to be the same

Region-based scoring functions can be used predict triples from the learned entity embeddings

Victor Charpenay, Steven Schockaert: Faithful Relational Reasoning with Region-based Embeddings: Expressivity of Convex Coordinate-wise Models. ICML 2026

Modelling rules with encoder-decoder models

Encoder: enforcing that all triples from the KG are satisfied w.r.t. some region-based relation embedding η

Decoder: verifying which other triples are satisfied w.r.t. some region-based relation embedding η'

GNN encoder: message-passing updates can sometimes be interpreted as gradient-based optimisation steps of **region-based relation embeddings model**

These region-based embedding models don't need to be the same

Region-based scoring functions can be used predict triples from the learned entity embeddings

If the encoder and decoder step are allowed to use different regions, then for any set of closed path rules P, there exists a coordinate-wise convex region-based model that captures exactly the set of rules that can be inferred in m steps from P

Takeaway messages

Modelling relationships between entities using neural representations can be done in two ways:

- Explicitly encoding, for each entity pair, how they are related
  - Difficult to do this in an exhaustive and scalable way (e.g. edge transformers need to compute a cubic number of relation compositions in each step)
- Encoding relationships implicitly through the relative position of entity embeddings
  - Can be efficient, although “practical” modelling choices come with unavoidable limitations on expressivity

Takeaway messages

Modelling relationships between entities using neural representations can be done in two ways:

- Explicitly encoding, for each entity pair, how they are related
  - Difficult to do this in an exhaustive and scalable way (e.g. edge transformers need to compute a cubic number of relation compositions in each step)
- Encoding relationships implicitly through the relative position of entity embeddings
  - Can be efficient, although “practical” modelling choices come with unavoidable limitations on expressivity

Optimal approach may require a combination of both strategies

- LLMs appear to model relations, in part, based on relative positions between entity embeddings
- But other mechanisms have also been uncovered

Takeaway messages

Modelling relationships between entities using neural representations can be done in two ways:

- Explicitly encoding, for each entity pair, how they are related
  - Difficult to do this in an exhaustive and scalable way (e.g. edge transformers need to compute a cubic number of relation compositions in each step)
- Encoding relationships implicitly through the relative position of entity embeddings
  - Can be efficient, although “practical” modelling choices come with unavoidable limitations on expressivity

Optimal approach may require a combination of both strategies

- LLMs appear to model relations, in part, based on relative positions between entity embeddings
- But other mechanisms have also been uncovered

Our understanding of how different kinds of knowledge can be faithfully encoded using vectors and neural models is still very limited

- This matters for the design of more robust AI systems, going beyond chain-of-thought based LLM reasoners, and for clarifying the role of symbolic reasoning within neuro-symbolic systems

Open problems

Designing more expressive models

- Studying other kinds of rules (e.g. rules with unary predicates, intersection rules)
- Disjunctive rules (e.g. for reasoning with ambiguous knowledge)
- Capturing defeasible/probabilistic rule bases

Open problems

Designing more expressive models

- Studying other kinds of rules (e.g. rules with unary predicates, intersection rules)
- Disjunctive rules (e.g. for reasoning with ambiguous knowledge)
- Capturing defeasible/probabilistic rule bases

Theoretical constructions typically require unrealistically high dimensionality and sometimes rely on “unstable” constructions (where very small changes in coordinates can have a big impact)

- Can we study which types of models are best suited for approximating knowledge bases in low-dimensional spaces?
- Can we study how knowledge can be encoded using “robust” models?

Open problems

Designing more expressive models

- Studying other kinds of rules (e.g. rules with unary predicates, intersection rules)
- Disjunctive rules (e.g. for reasoning with ambiguous knowledge)
- Capturing defeasible/probabilistic rule bases

Theoretical constructions typically require unrealistically high dimensionality and sometimes rely on “unstable” constructions (where very small changes in coordinates can have a big impact)

- Can we study which types of models are best suited for approximating knowledge bases in low-dimensional spaces?
- Can we study how knowledge can be encoded using “robust” models?

Improving empirical performance

- Stronger inductive biases are needed (TransE can learn relation embeddings from a single example)
- We need better benchmarks for evaluating reasoning abilities