Lecture 2: Knowledge Graph Embedding Models

Relational Learning

İsmail İlkan Ceylan

Advanced Topics in Machine Learning, University of Oxford

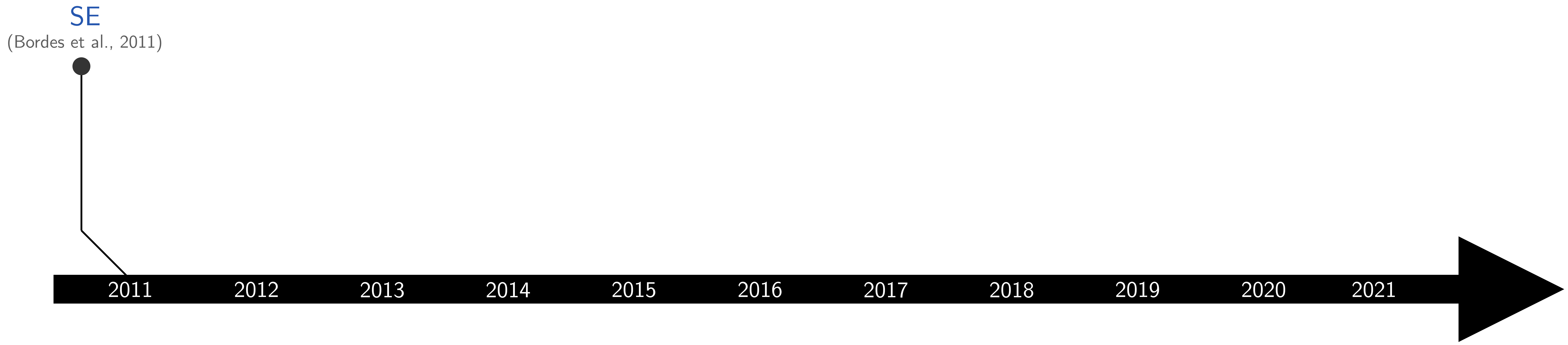
17.01.2022

Overview

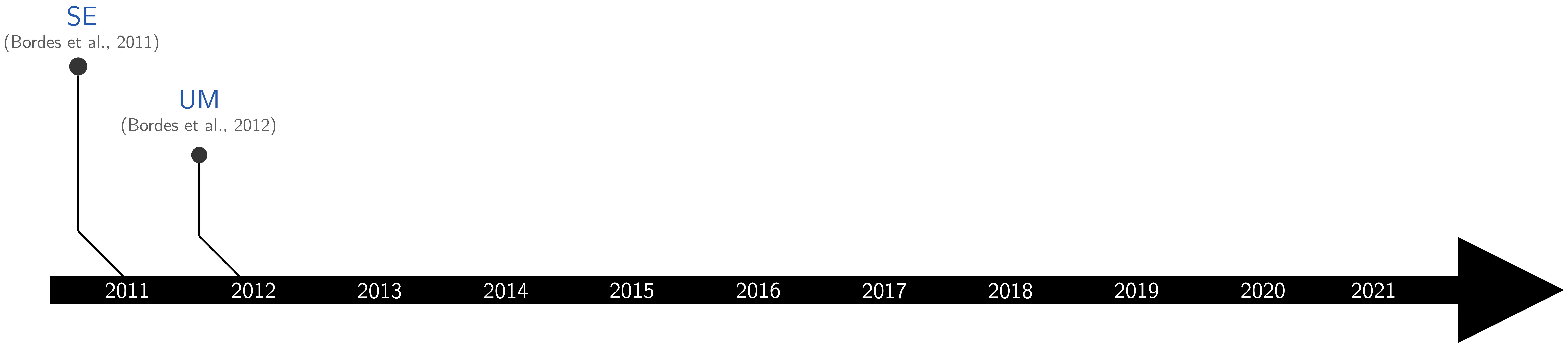
- A glimpse at embedding models
  - Translational models: TransE and RotatE
  - Bilinear models: RESCAL, DistMult, and ComplEx
  - Box embedding models
- Overview of the embedding models
- Temporal knowledge graph completion
- Outlook and discussions
- Summary

A Glimpse at Embedding Models

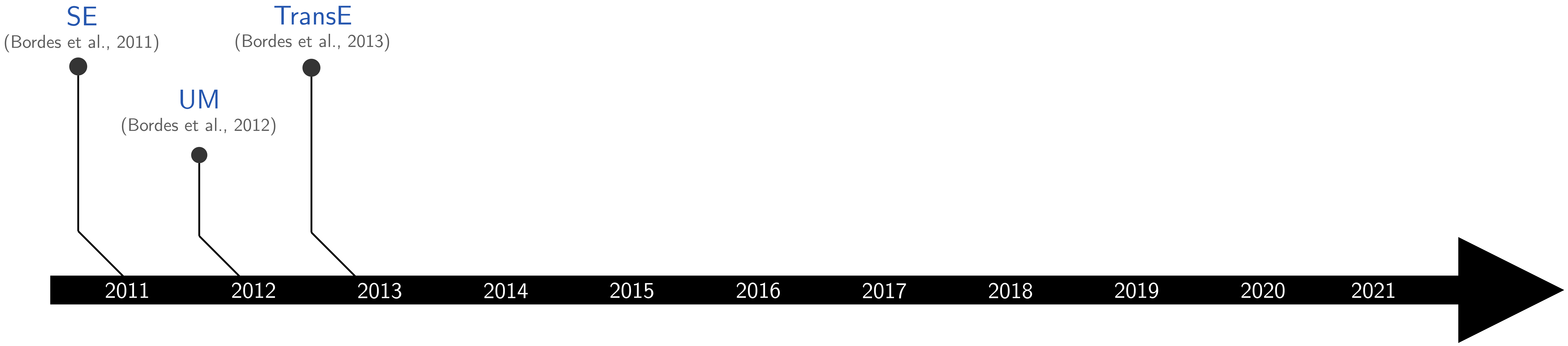
A Glimpse at Embedding Models



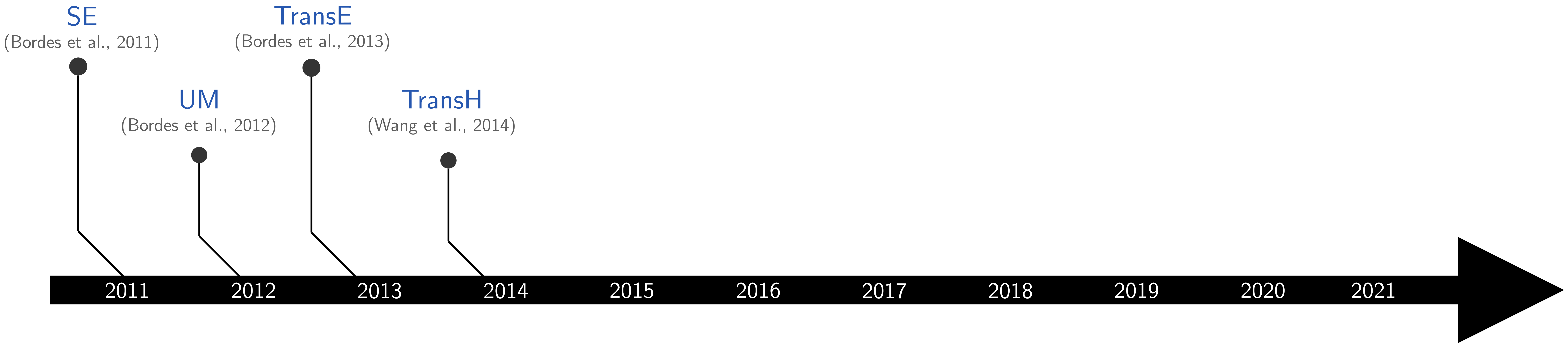
A Glimpse at Embedding Models



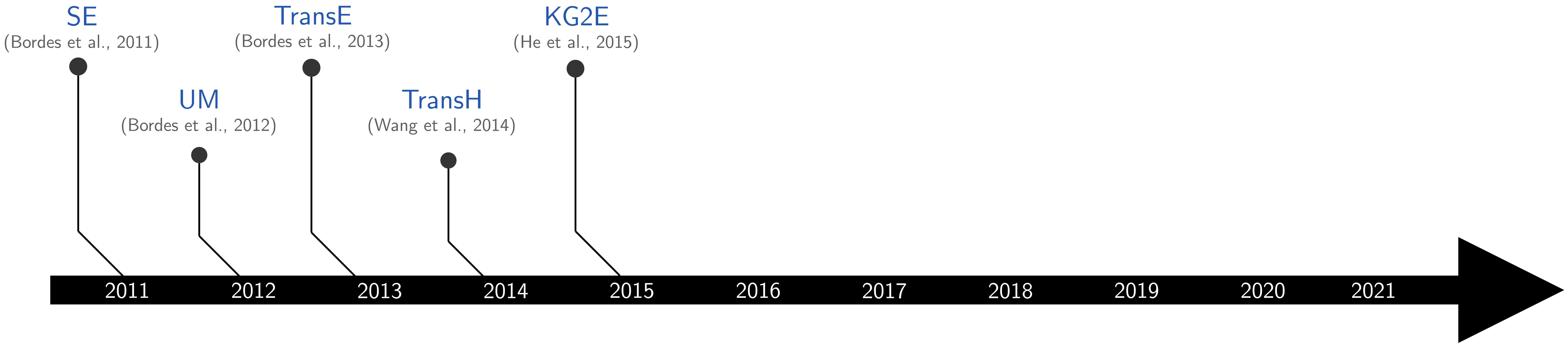
A Glimpse at Embedding Models



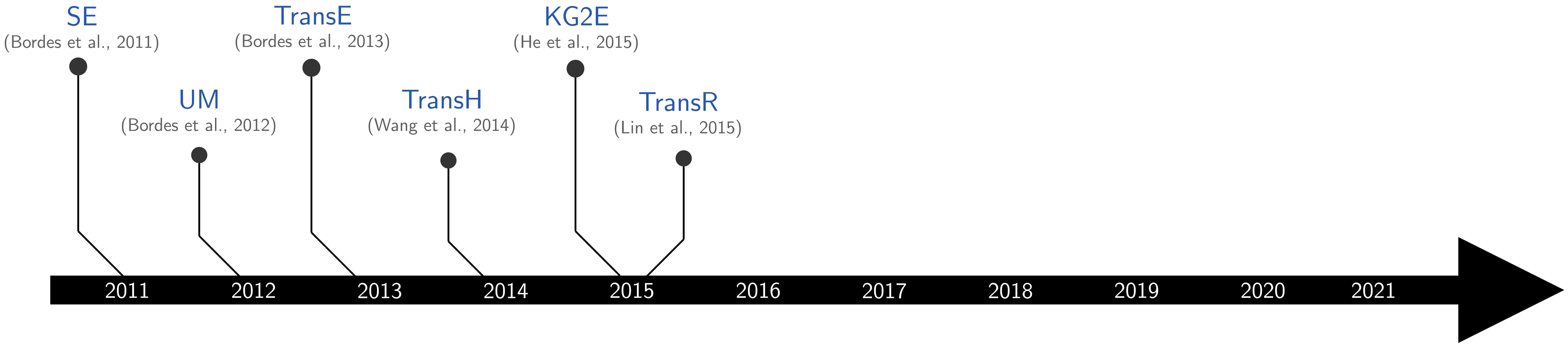
A Glimpse at Embedding Models



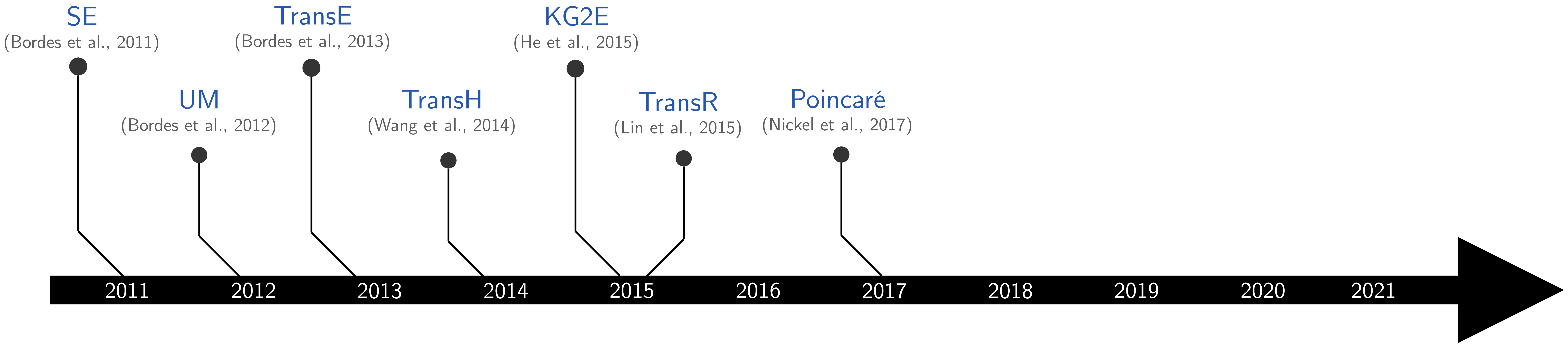
A Glimpse at Embedding Models



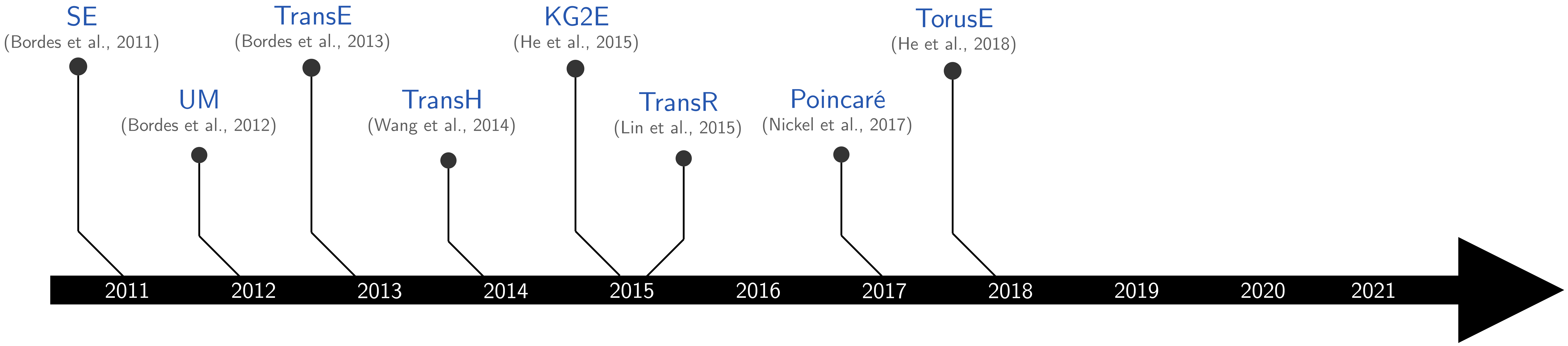
A Glimpse at Embedding Models



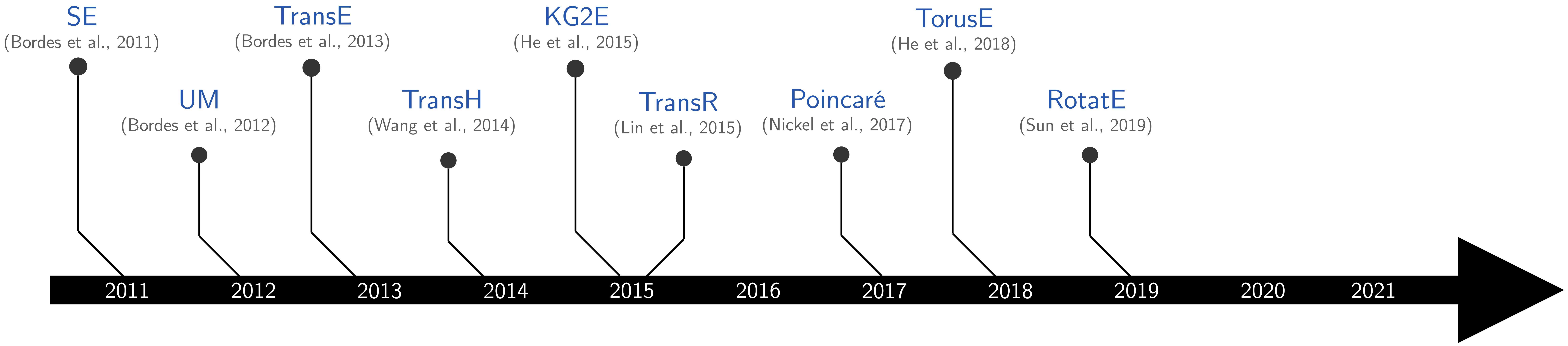
A Glimpse at Embedding Models



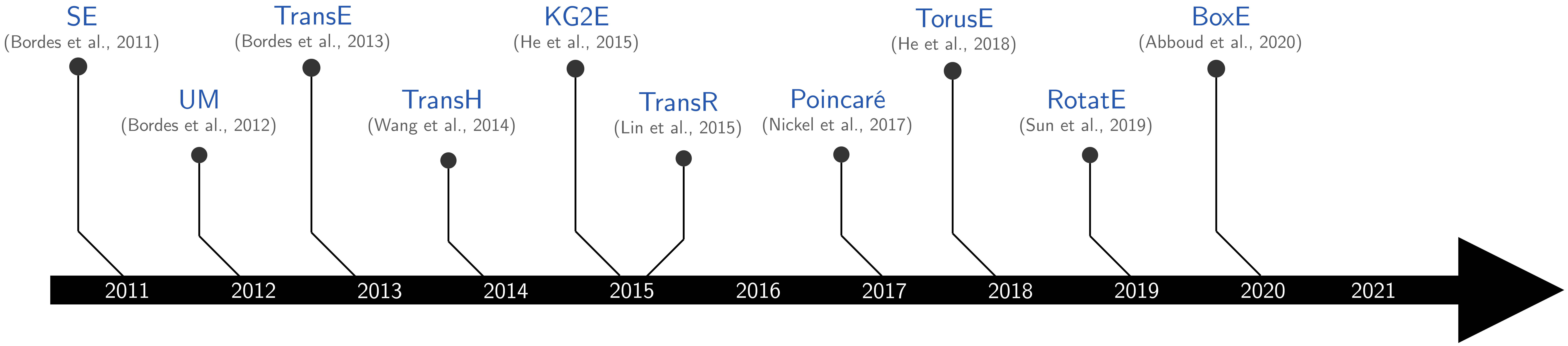
A Glimpse at Embedding Models



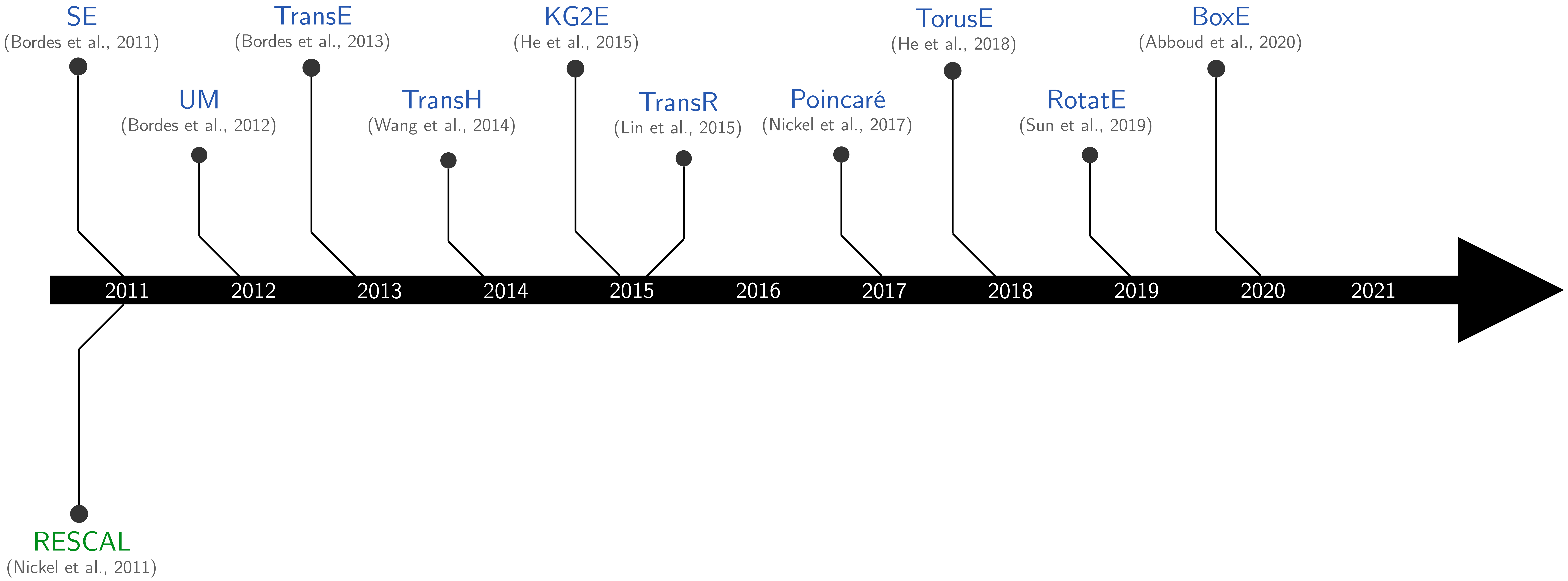
A Glimpse at Embedding Models



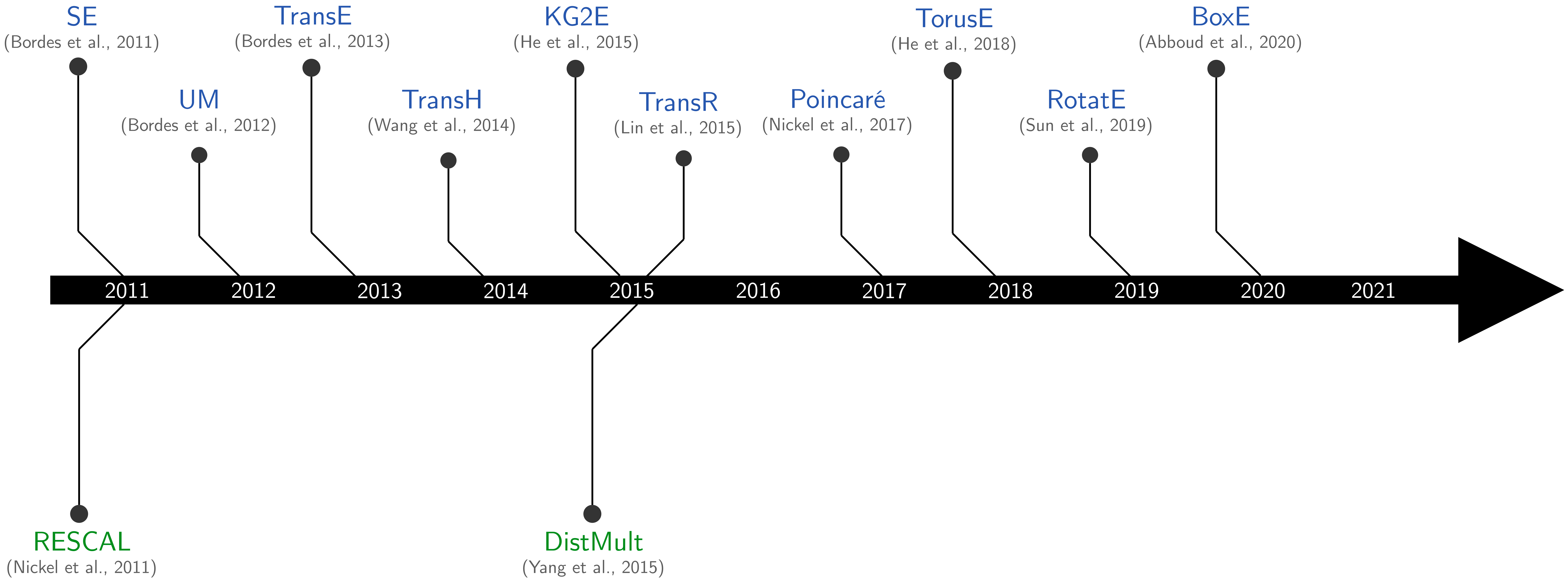
A Glimpse at Embedding Models



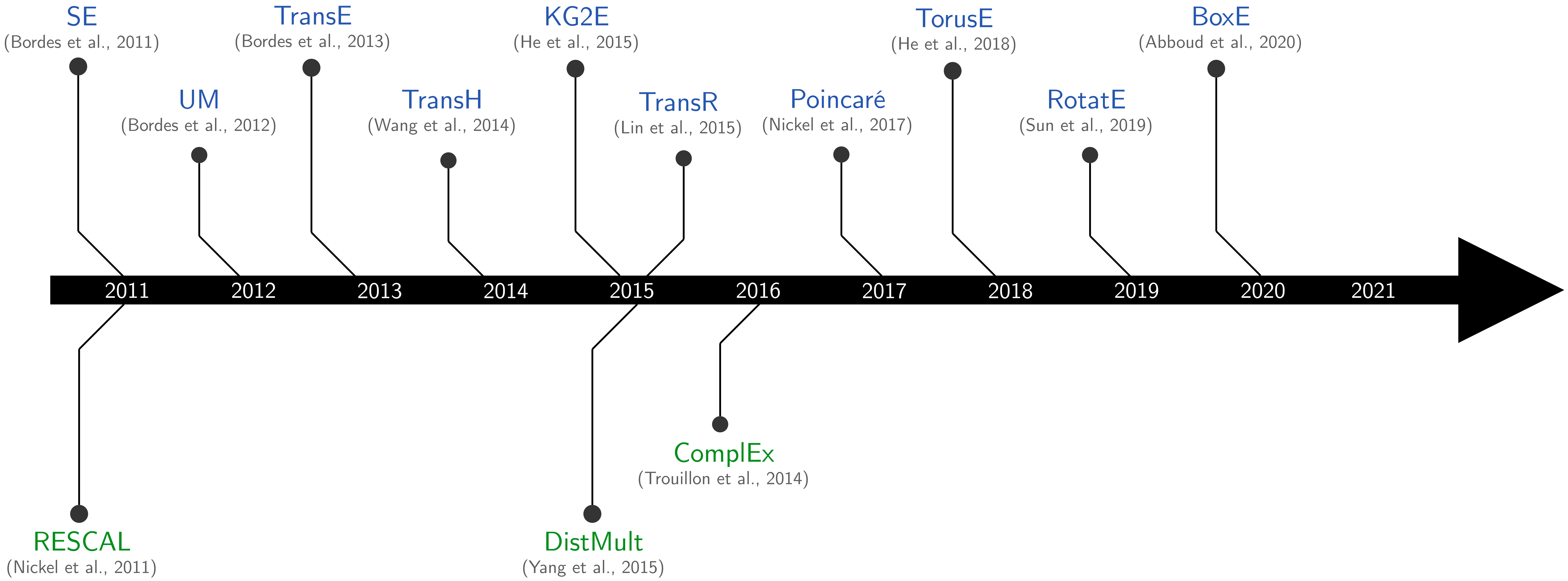
A Glimpse at Embedding Models



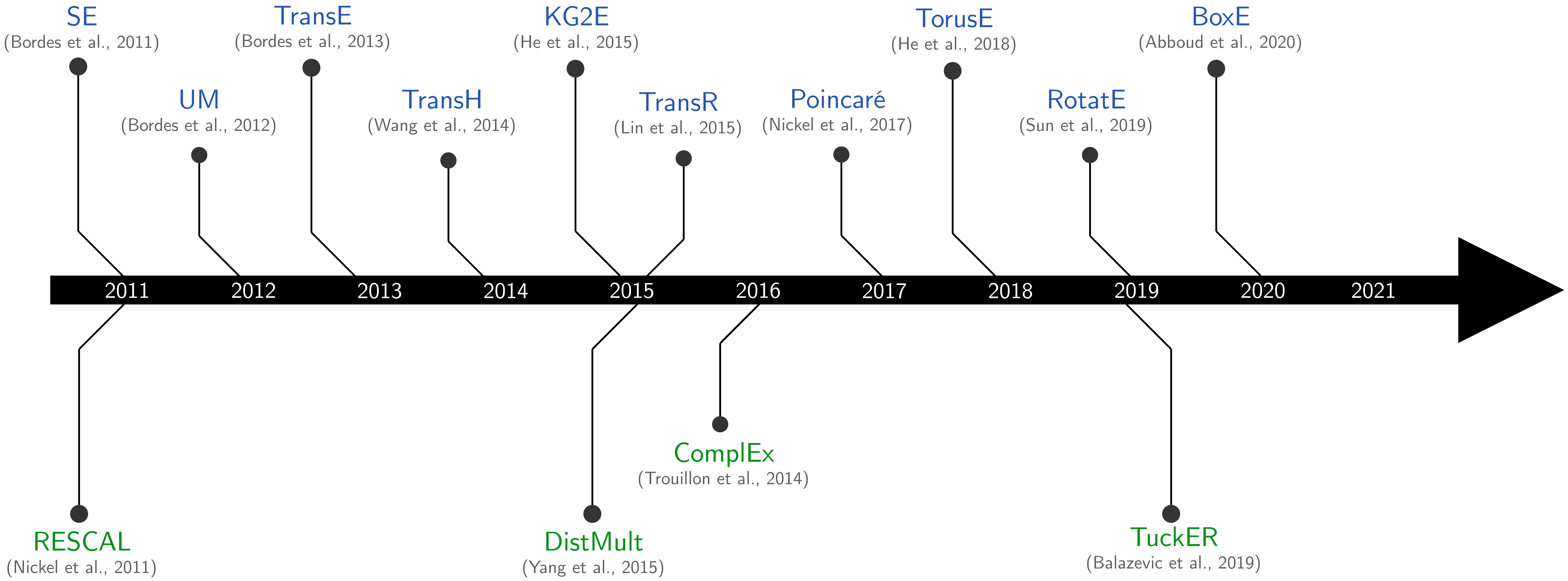
A Glimpse at Embedding Models



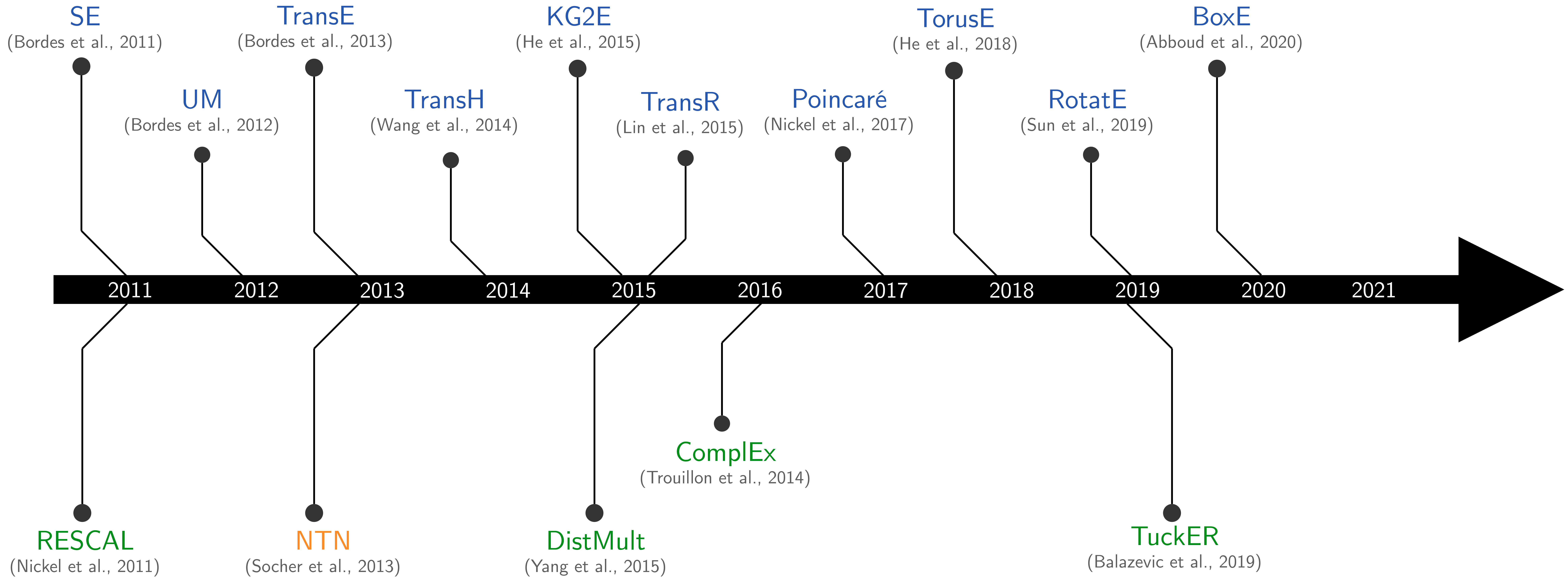
A Glimpse at Embedding Models



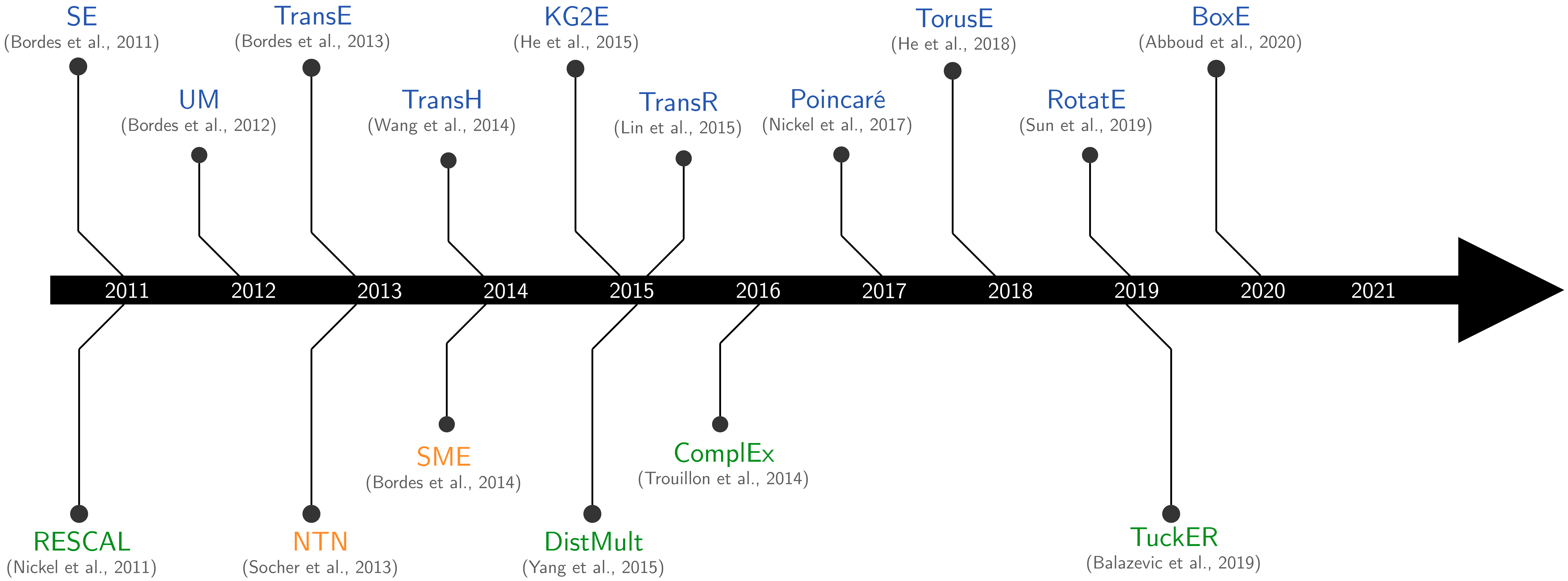
A Glimpse at Embedding Models



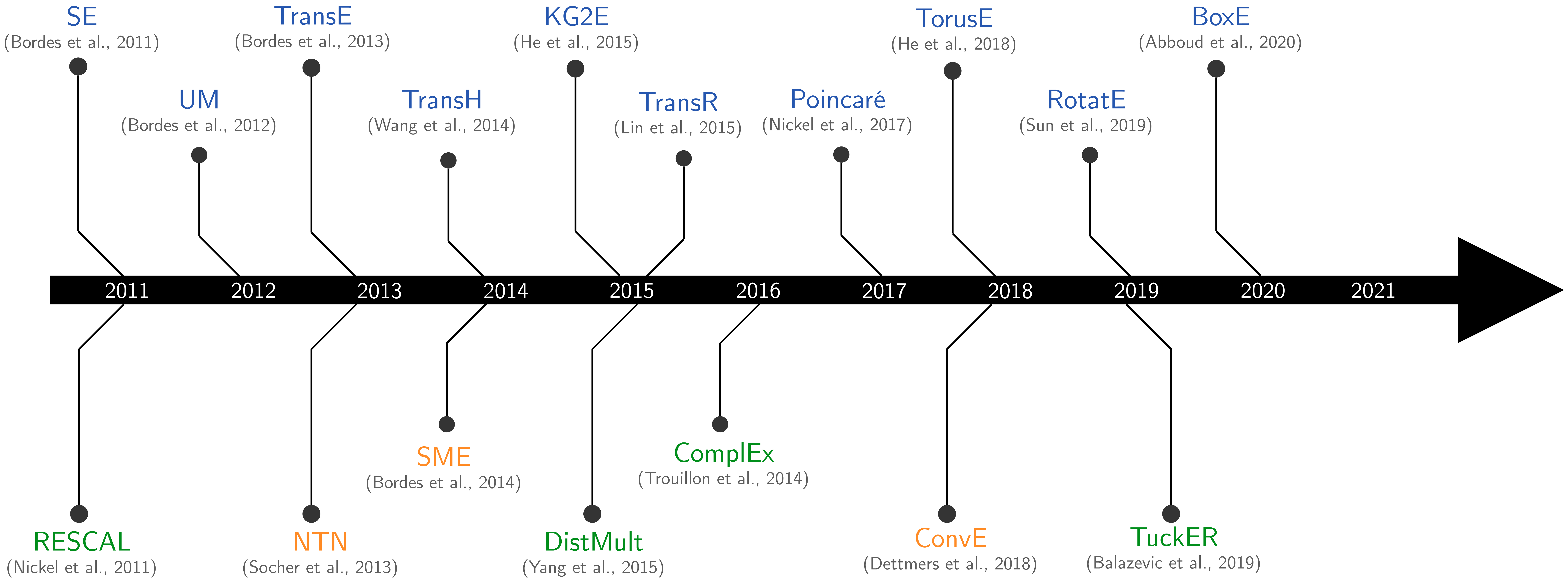
A Glimpse at Embedding Models



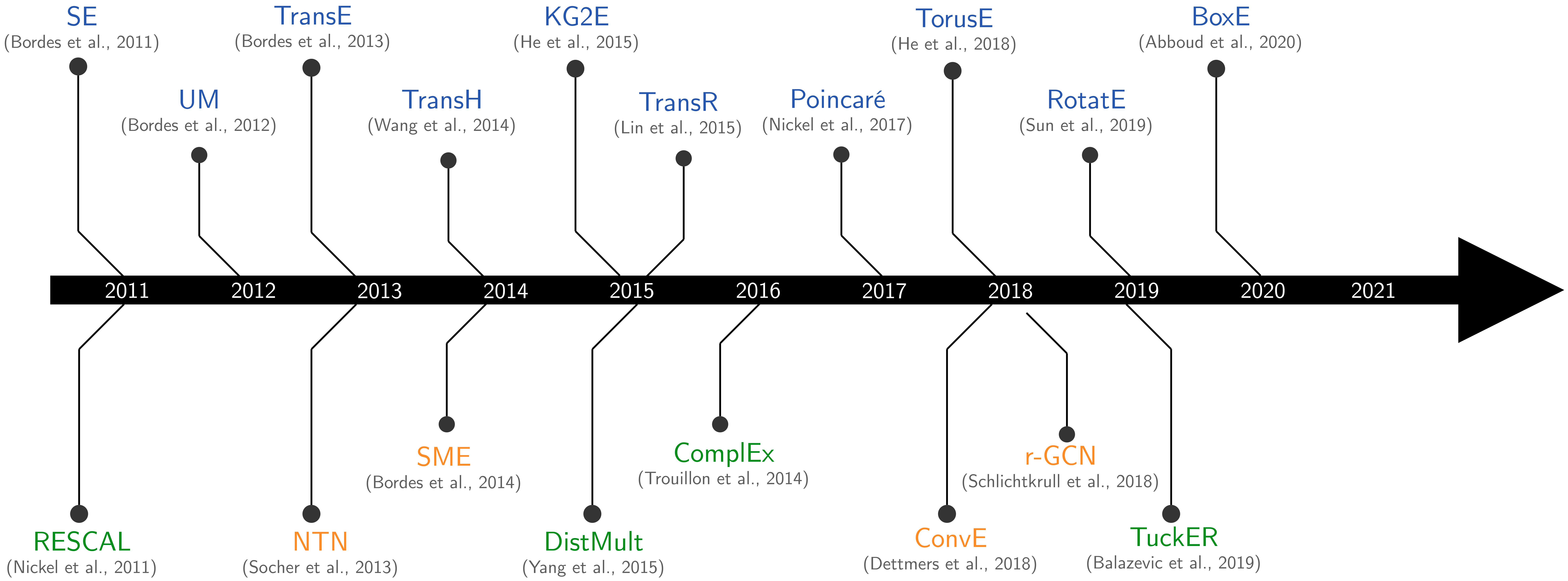
A Glimpse at Embedding Models



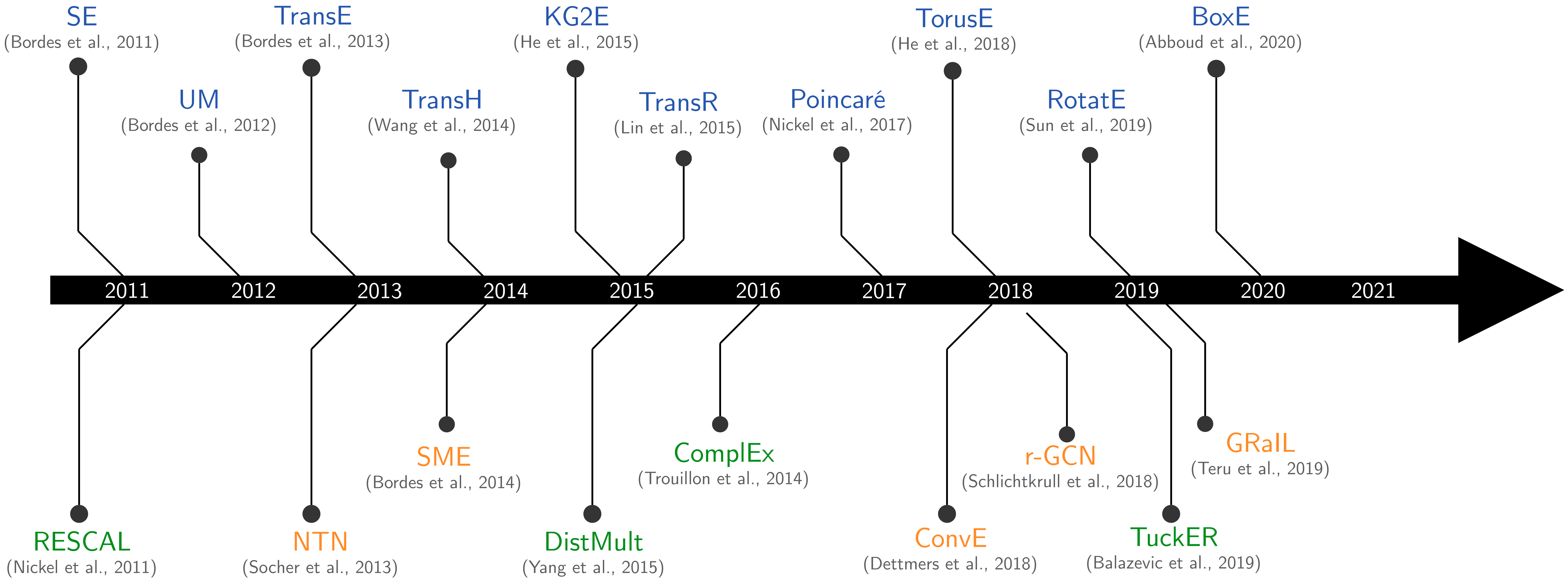
A Glimpse at Embedding Models



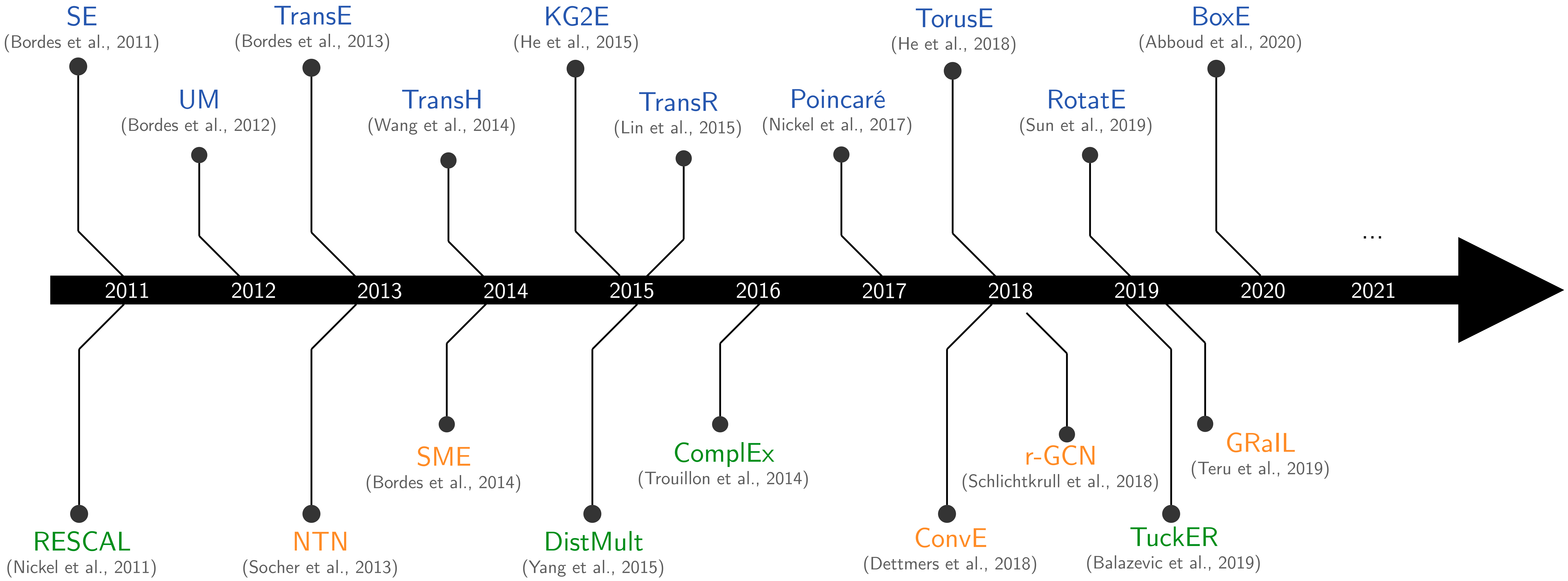
A Glimpse at Embedding Models



A Glimpse at Embedding Models



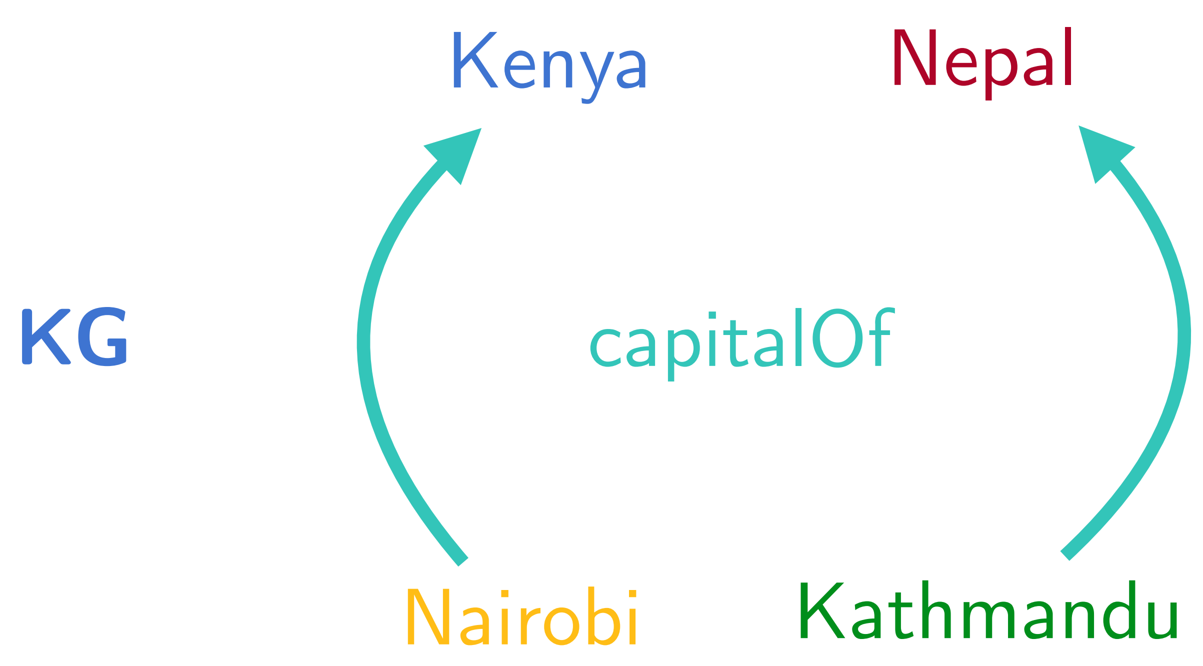
A Glimpse at Embedding Models



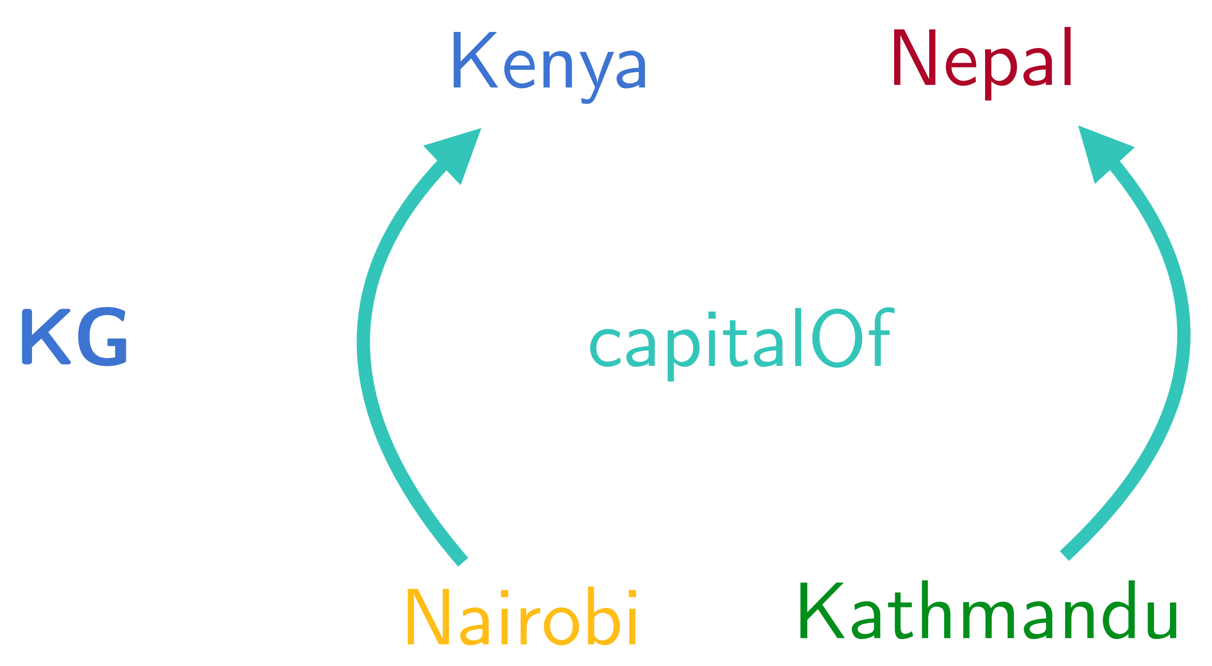
Translational Models

TransE

TransE

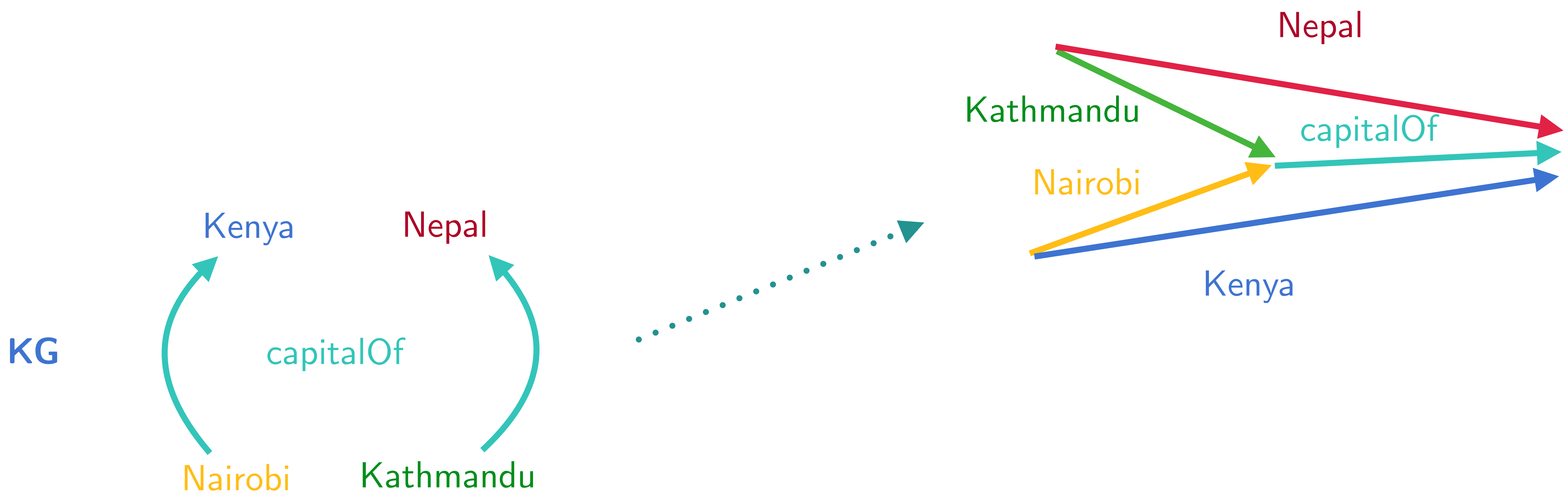


TransE



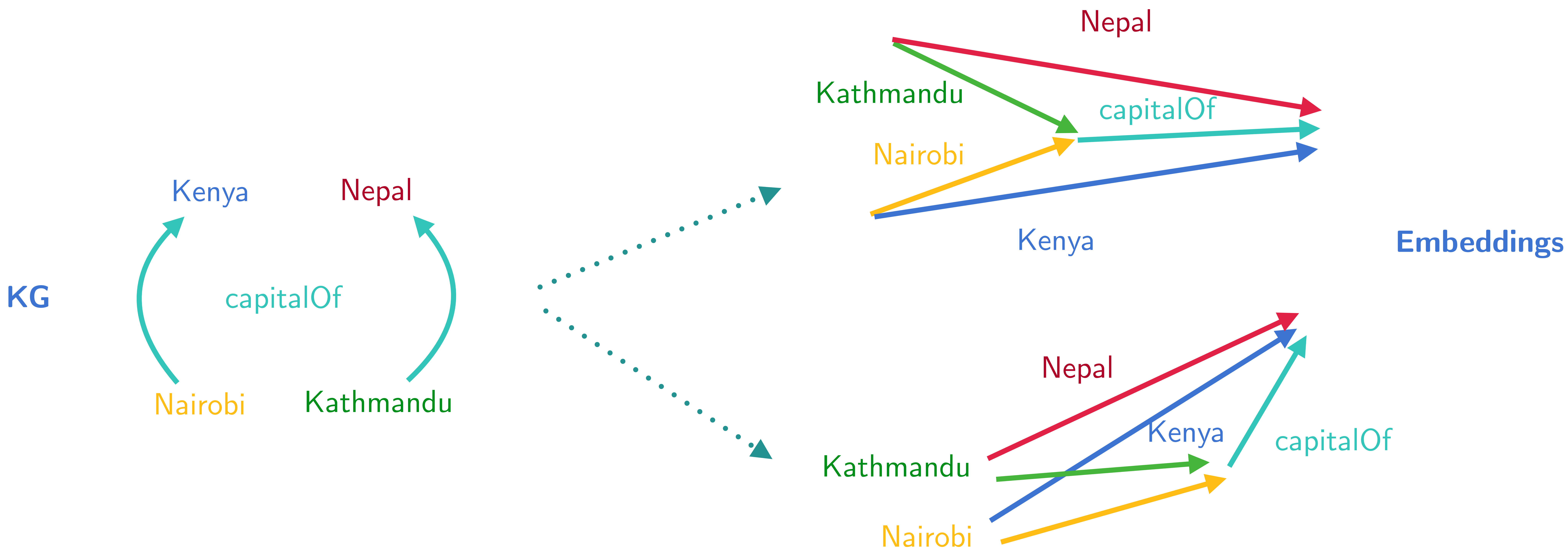
- **Encoder**: Represents entities $h, t \in E$ and relations $r \in R$, through d-dimensional vectors $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$.

TransE



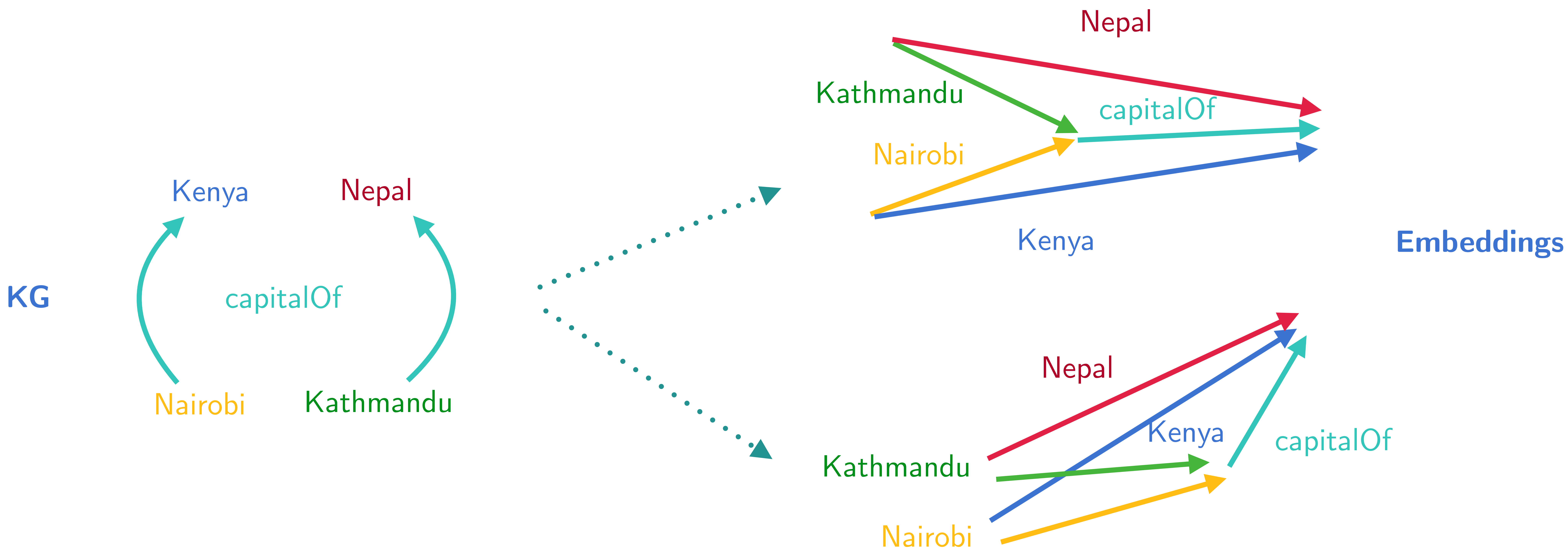
- **Encoder**: Represents entities $h, t \in E$ and relations $r \in R$, through d-dimensional vectors $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$.

TransE



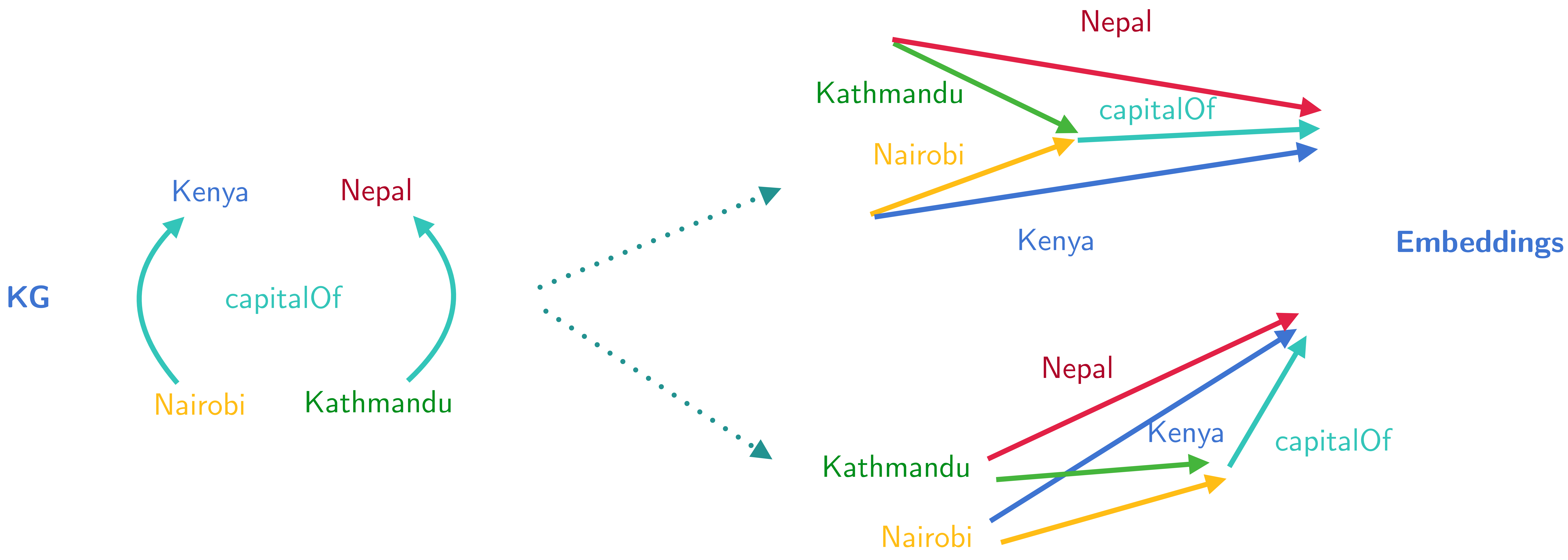
- **Encoder**: Represents entities $h, t \in E$ and relations $r \in R$, through d-dimensional vectors $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$.

TransE



- **Encoder**: Represents entities $h, t \in E$ and relations $r \in R$, through d-dimensional vectors $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$.
- **Decoder**: Scores a fact $r(h, t)$ depending how similar $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$ are, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$.

TransE



- **Encoder**: Represents entities $h, t \in E$ and relations $r \in R$, through d-dimensional vectors $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$.
- **Decoder**: Scores a fact $r(h, t)$ depending how similar $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$ are, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$.
- Optimized to minimize (resp., maximize) the dissimilarity of true facts (resp., negative facts).

TransE: Optimization, Loss, Training

Decoder: Consider a distance measure d, e.g., L_1 or L_2 norm, where $d(\mathbf{h} + \mathbf{r}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$ represents how dissimilar $\mathbf{h} + \mathbf{r}$, and $\mathbf{t}$ are. Hence, $-\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$ defines a similarity measure.

Loss: TransE defines the loss function:

$$\mathscr{L} = \sum_{r(h,t) \in G} \sum_{r(h',t') \in N^{r(h,t)}} \left[d(\mathbf{h} + \mathbf{r}, \mathbf{t}) - d(\mathbf{h}' + \mathbf{r}, \mathbf{t}') + \gamma \right]_+ ,$$

where γ is a margin hyper-parameter, $N^{r(h,t)}$ is a set of negative samples for $r(h,t)$, and $[x]_+$ denotes the positive part of x.

Favors higher values of similarity for true facts than for negative ones: implementation of the intended criterion.

Optimization: By stochastic gradient descent, where all embeddings are initialized randomly; at each iteration, the parameters are updated by taking a gradient step with constant learning rate. The algorithm is stopped based on its performance on a validation set.

How Expressive is TransE?

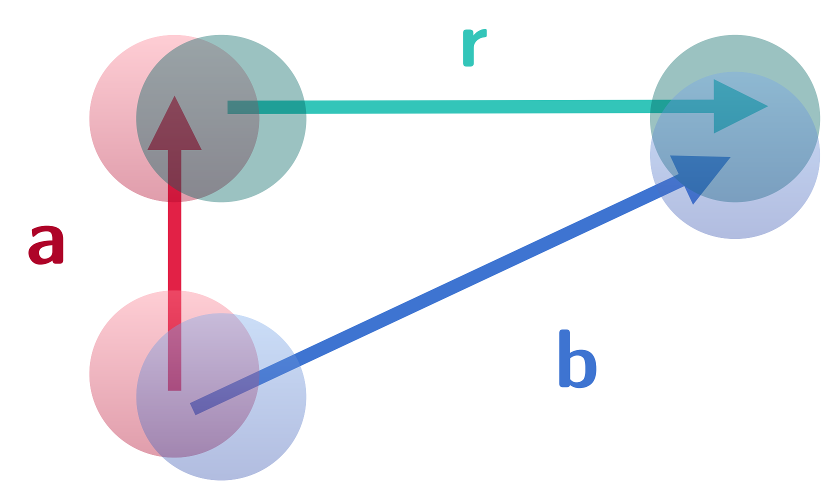
How Expressive is TransE?

Let us realize the set of true facts $\{r(a, b), r(b, a)\}$ in TransE. These facts can be clearly be realized independently, e.g., (i) & (ii).

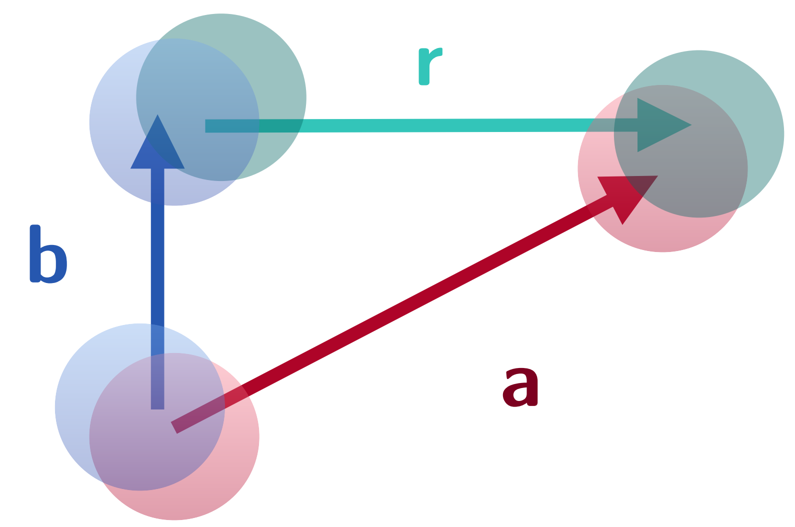
How Expressive is TransE?

Let us realize the set of true facts $\{r(a,b), r(b,a)\}$ in TransE. These facts can be clearly be realized independently, e.g., (i) & (ii).

(i) $r(a,b)$



(ii) $r(b,a)$

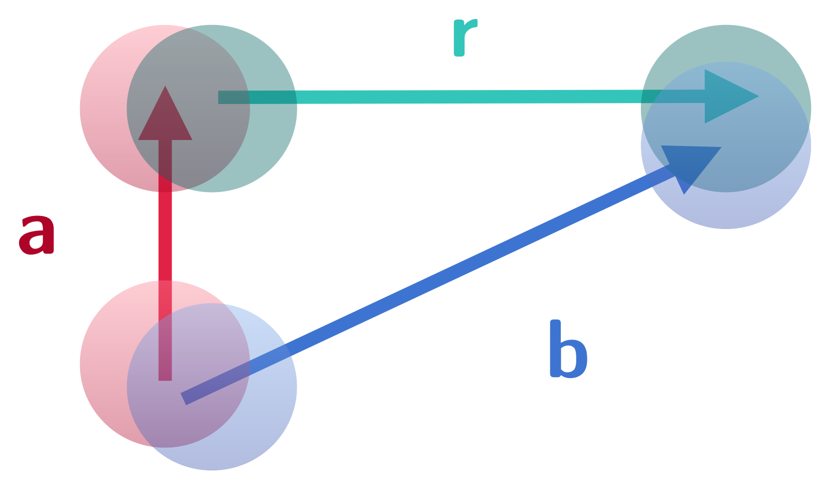


How Expressive is TransE?

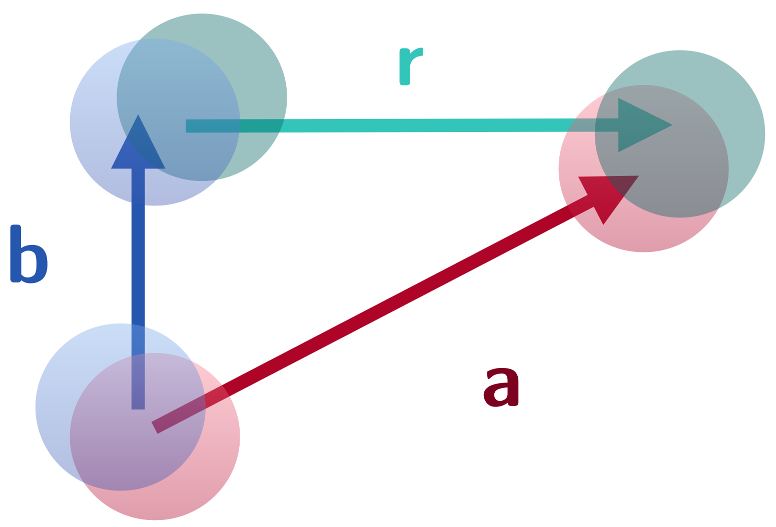
Let us realize the set of true facts $\{r(a,b), r(b,a)\}$ in TransE. These facts can be clearly be realized independently, e.g., (i) & (ii).

To realize these facts jointly, we need $r = 0$, as shown in (iii), but then $\{r(a,a), r(b,b)\}$ are true facts, although these could be false.

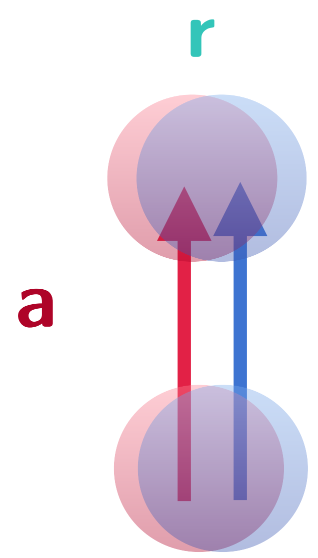
(i) $r(a,b)$



(ii) $r(b,a)$



(iii) $r(a,b)$ & $r(b,a)$



How Expressive is TransE?

Let us realize the set of true facts $\{r(a,b), r(b,a)\}$ in TransE. These facts can be clearly be realized independently, e.g., (i) & (ii).

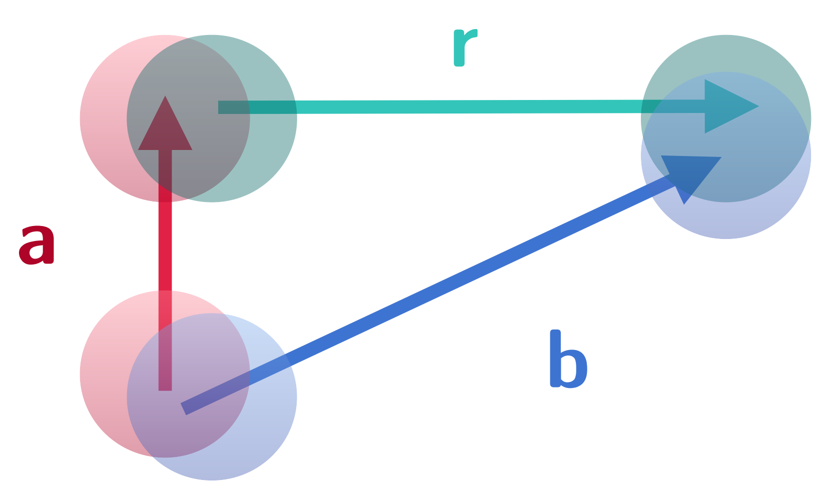
To realize these facts jointly, we need $r = 0$, as shown in (iii), but then $\{r(a,a), r(b,b)\}$ are true facts, although these could be false.

The relation r can be made symmetric only by additionally forcing r to be reflexive, hence leading to loss of generality.

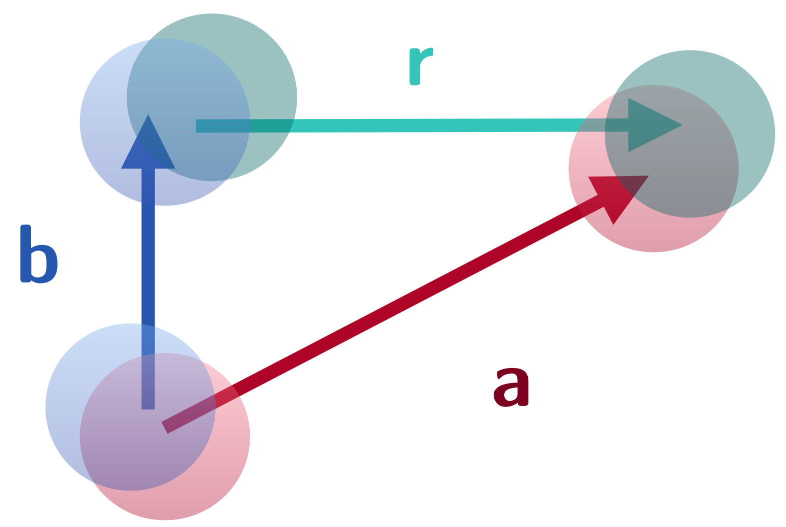
TransE is not fully expressive: it cannot encode the set of true facts $\{r(a,b), r(b,a)\}$ and the set of false facts $\{r(a,a), r(b,b)\}$.

Consider a relation such as cousinOf with entities alice, bob to see a problematic example.

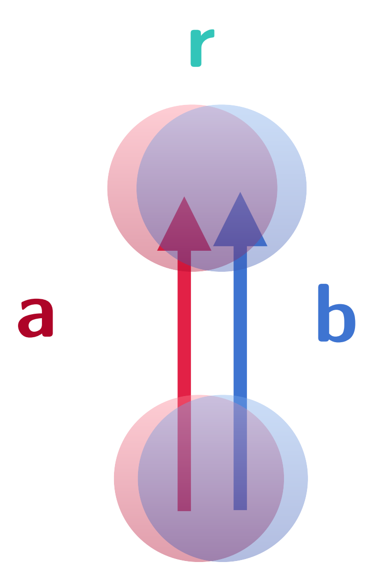
(i) $r(a,b)$



(ii) $r(b,a)$



(iii) $r(a,b)$ & $r(b,a)$

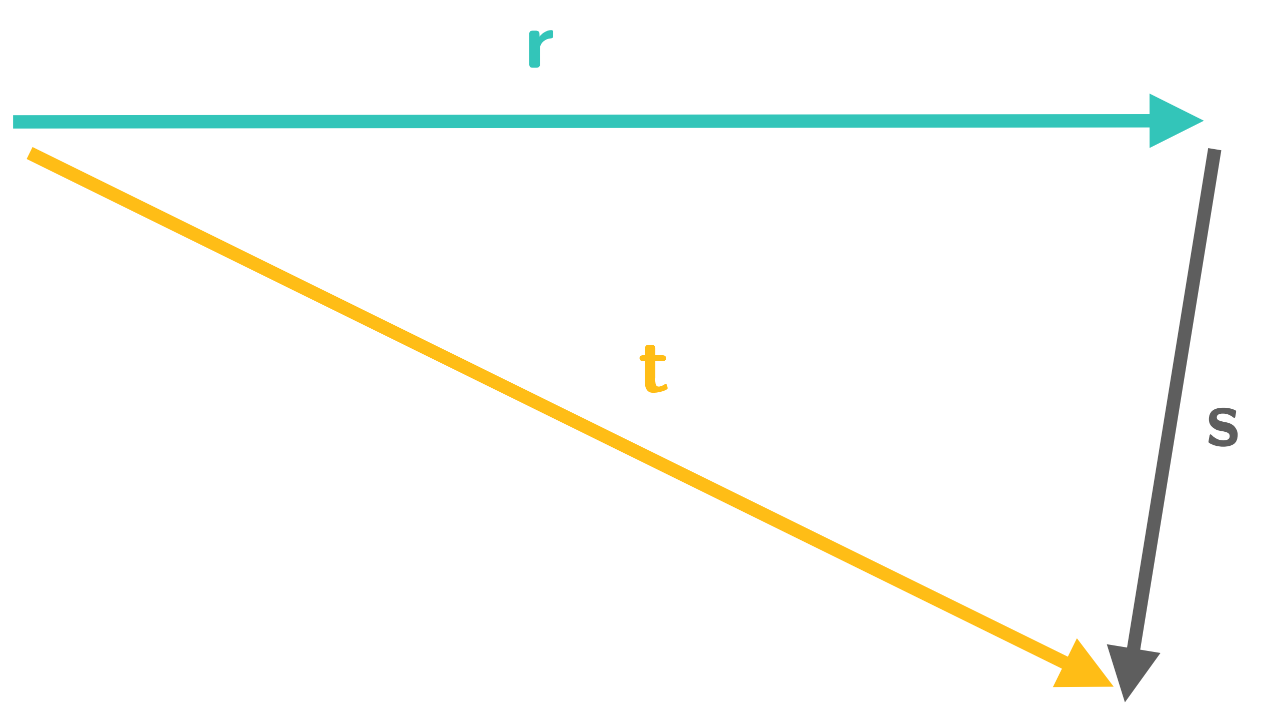


Which Inference Patterns Can TransE Capture?

Which Inference Patterns Can TransE Capture?

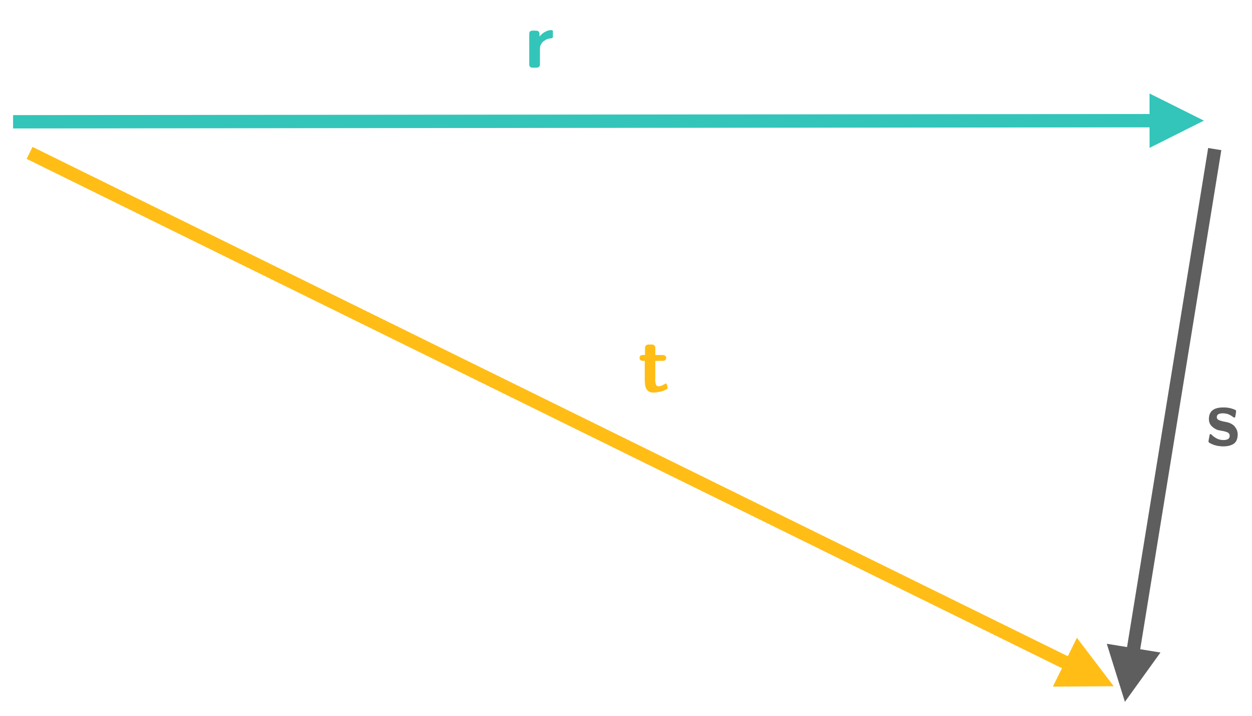
Can TransE capture the composition pattern: $\forall x, y, z \;\; r(x, y) \wedge s(y, z) \Rightarrow t(x, z)$?

Which Inference Patterns Can TransE Capture?



Can TransE capture the composition pattern: $\forall x, y, z \; r(x, y) \wedge s(y, z) \Rightarrow t(x, z)$?

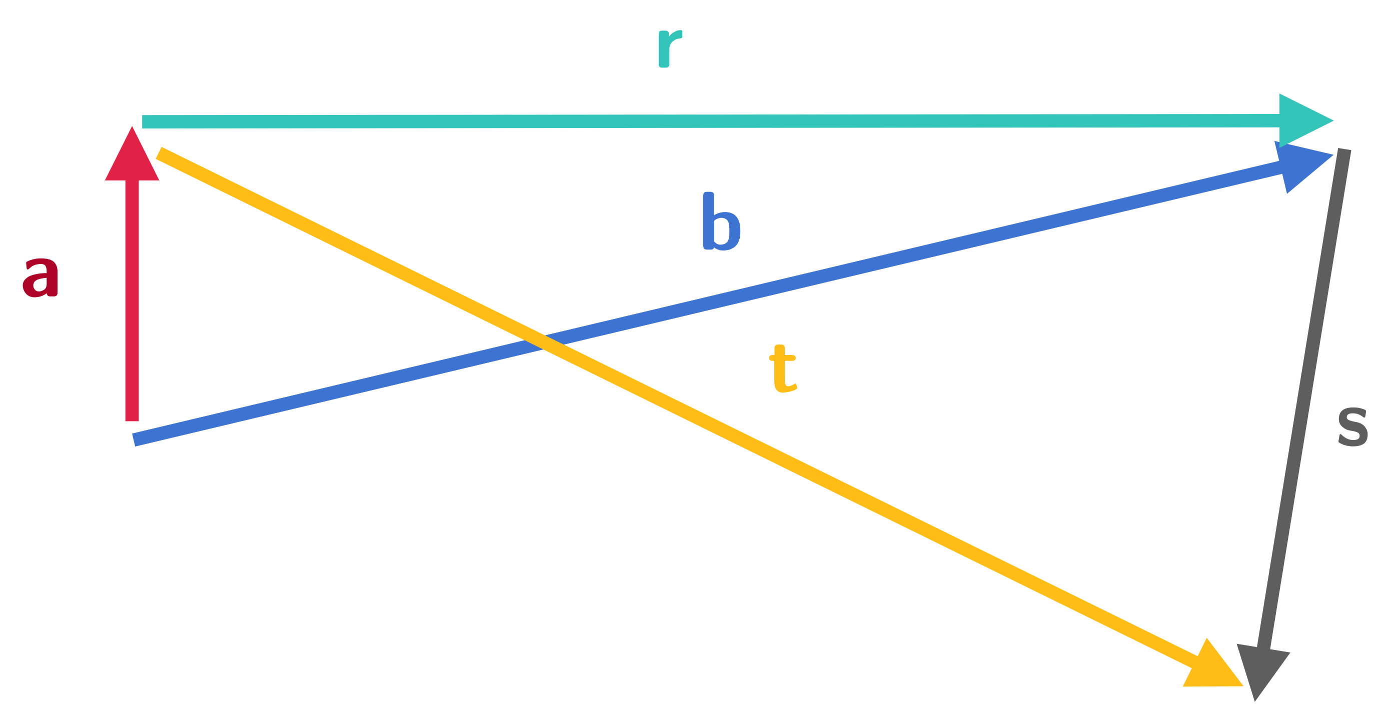
Which Inference Patterns Can TransE Capture?



Can TransE capture the composition pattern: $\forall x, y, z \; r(x, y) \wedge s(y, z) \Rightarrow t(x, z)$?

Configure the model as $\mathbf{r} + \mathbf{s} \approx \mathbf{t}$: for any $a, b, c \in E$, whenever $r(a, b), s(b, c)$ hold, so does $t(a, c)$.

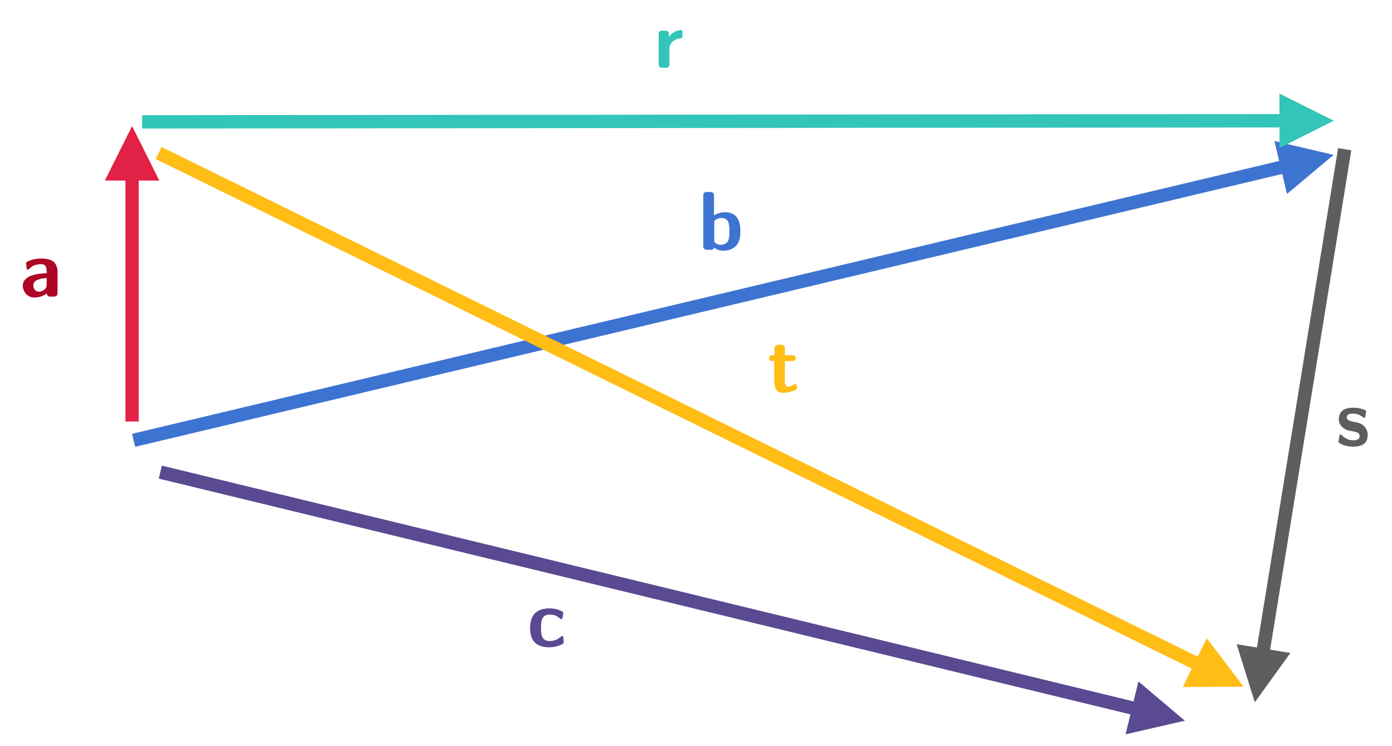
Which Inference Patterns Can TransE Capture?



Can TransE capture the composition pattern: $\forall x, y, z\ r(x, y) \wedge s(y, z) \Rightarrow t(x, z)$?

Configure the model as $\mathbf{r} + \mathbf{s} \approx \mathbf{t}$: for any $a, b, c \in E$, whenever $r(a, b), s(b, c)$ hold, so does $t(a, c)$.

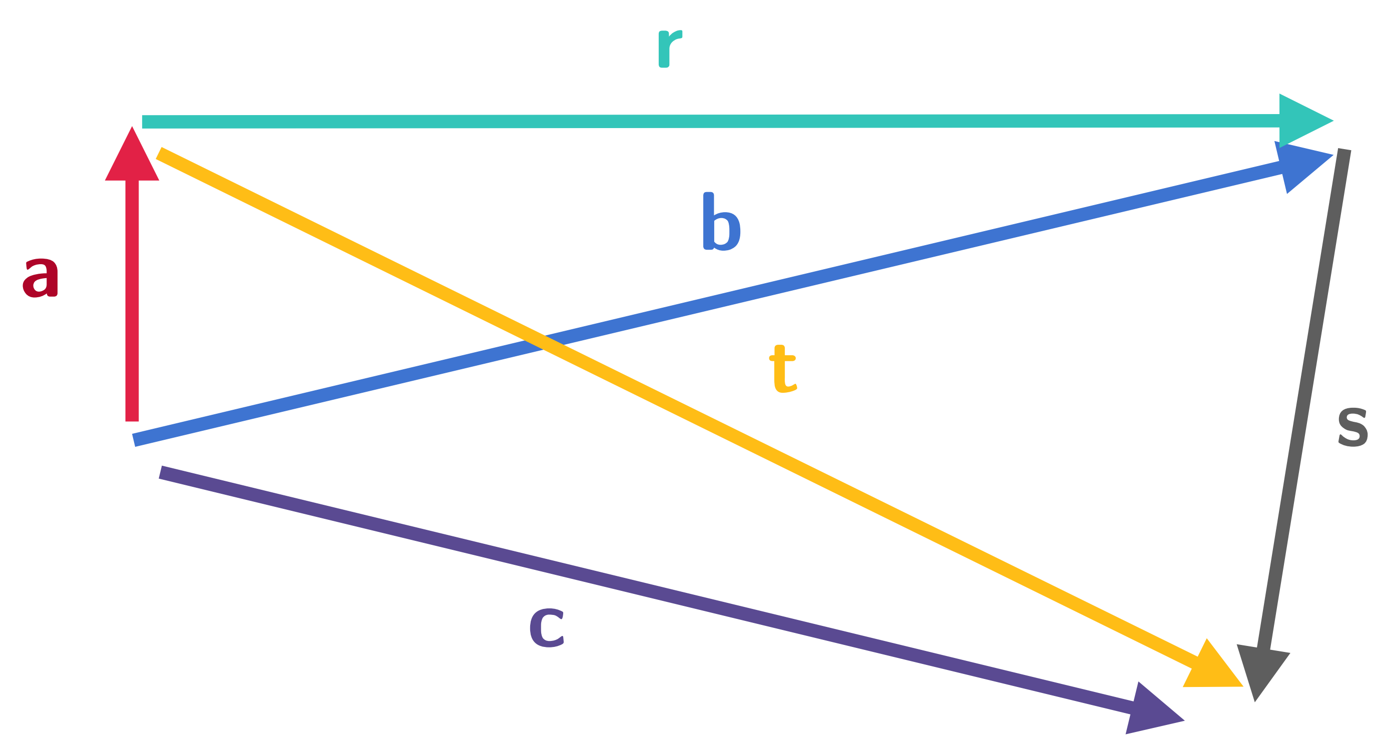
Which Inference Patterns Can TransE Capture?



Can TransE capture the composition pattern: $\forall x, y, z \; r(x, y) \wedge s(y, z) \Rightarrow t(x, z)$?

Configure the model as $\mathbf{r} + \mathbf{s} \approx \mathbf{t}$: for any $a, b, c \in E$, whenever $r(a, b), s(b, c)$ hold, so does $t(a, c)$.

Which Inference Patterns Can TransE Capture?

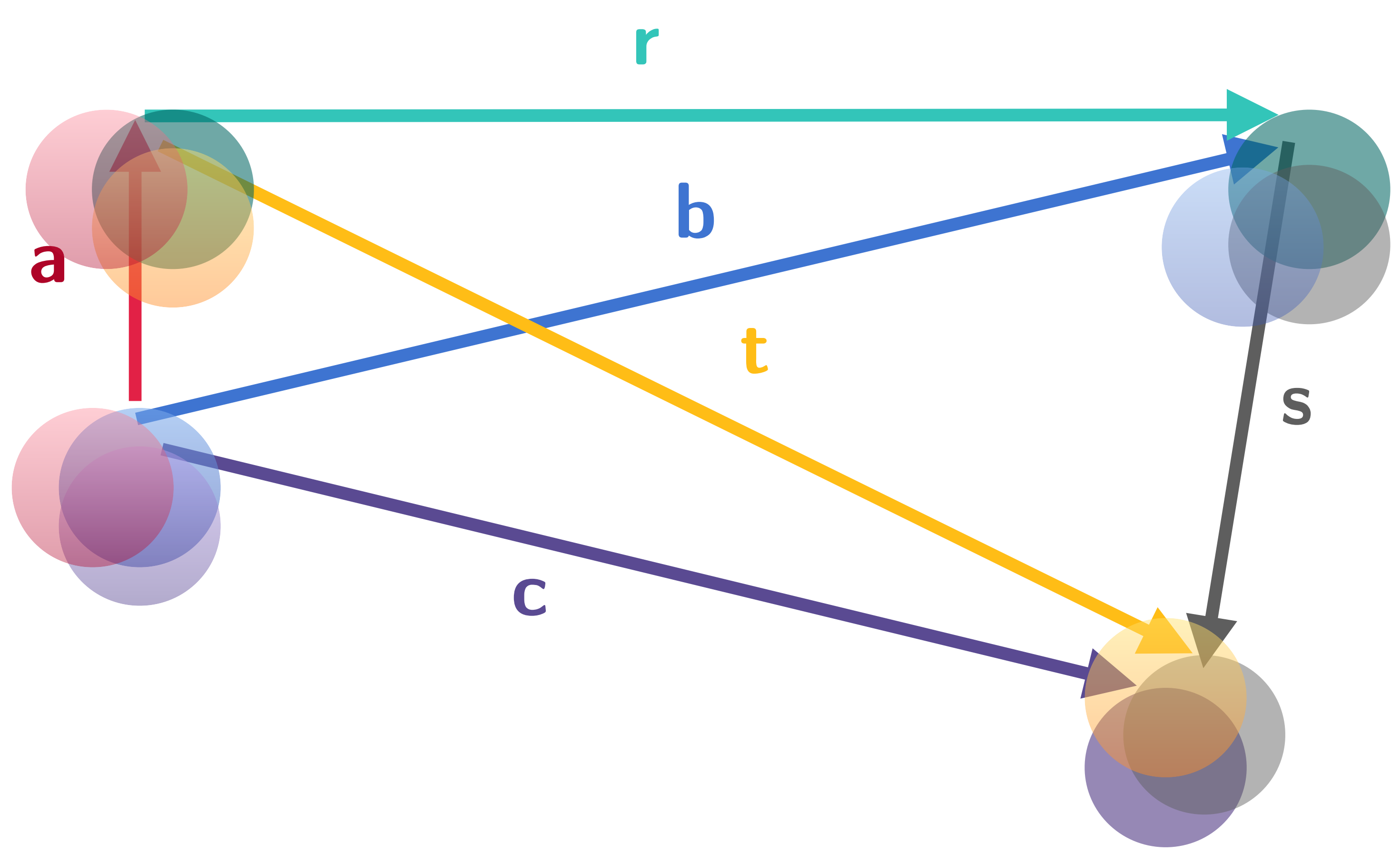


Can TransE capture the composition pattern: $\forall x, y, z\ r(x, y) \wedge s(y, z) \Rightarrow t(x, z)$?

Configure the model as $\mathbf{r} + \mathbf{s} \approx \mathbf{t}$: for any $a, b, c \in E$, whenever $r(a, b), s(b, c)$ hold, so does $t(a, c)$.

TransE can also capture, e.g., anti-symmetry and inversion. It can capture intersection only in a loose sense by tweaking the margins.

Which Inference Patterns Can TransE Capture?



Can TransE capture the composition pattern: $\forall x, y, z \; r(x, y) \wedge s(y, z) \Rightarrow t(x, z)$?

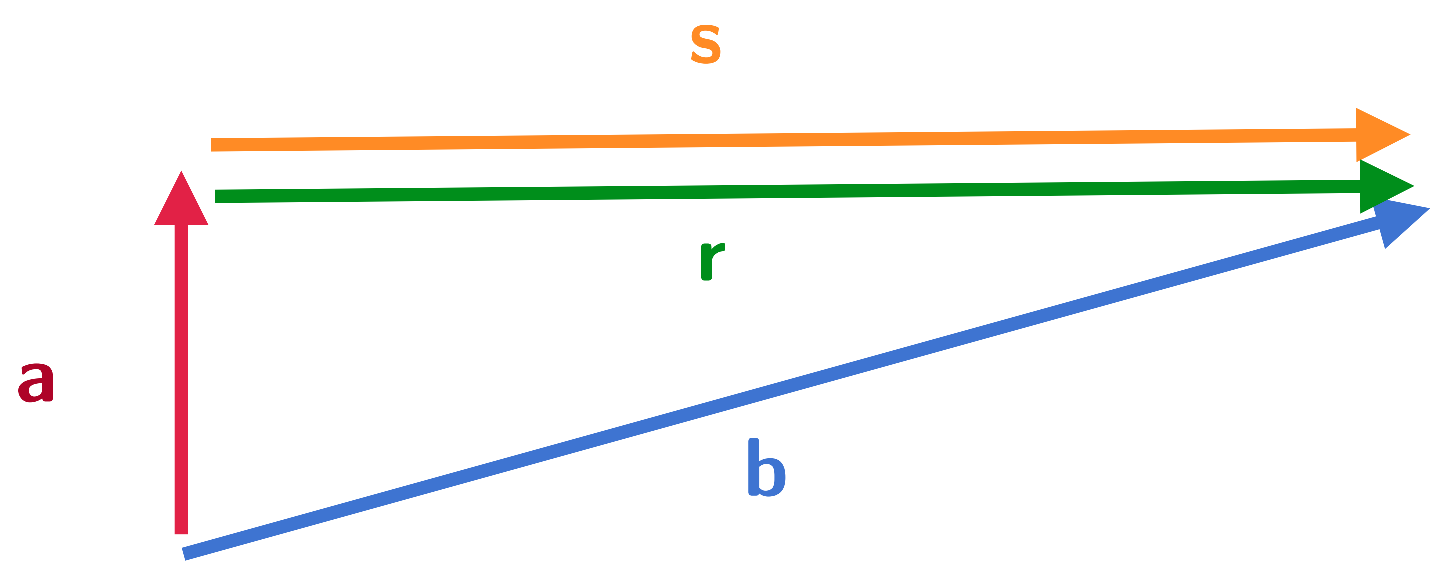
Configure the model as $\mathbf{r} + \mathbf{s} \approx \mathbf{t}$: for any $a, b, c \in E$, whenever $r(a, b), s(b, c)$ hold, so does $t(a, c)$.

TransE can also capture, e.g., anti-symmetry and inversion. It can capture intersection only in a loose sense by tweaking the margins.

Which Inference Patterns Can TransE Not Capture?

TransE cannot capture symmetry: a relation can be symmetric only by forcing it to be reflexive.

What about the hierarchy pattern: $\forall x, y\ r(x, y) \Rightarrow s(x, y)$?



Only by setting $\mathbf{r} \approx \mathbf{s}$, and, this would imply relation equivalence: TransE cannot capture hierarchy either.

The lack of ability to capture the hierarchy pattern is also a serious limitation, as it is also prevalent in datasets (e.g., the relation capitalOf implies the relation cityIn).

Other Limitations of TransE

1-to-n, n-to-1, n-to-n, relations refer to the cardinality of the relation in terms of the head and tail entities.

TransE does not efficiently learn the representations for n-to-n relations in a KG:

locatedIn(Oxford, Oxfordshire)

locatedIn(Oxford, UK)

We need Oxfordshire $\approx$ UK to realize these, since the elements locatedIn, Oxford are shared in the scoring.

Similarly for 1-to-n relations, i.e., Bob $\approx$ Chris in:

motherOf(Anne, Bob)

motherOf(Anne, Chris)

Other translational models are proposed to reduce the effect of this problem; see, e.g., TransH and TransR.

RotatE

RotatE

RotatE defines each relation r as a rotation from an entity h to an entity t in the complex vector space.

Euler's formula: $e^{i\theta} = cos\theta + i\ sin\theta$, i.e., a unitary complex number tracing the unit circle in the complex plane as θ ranges through reals.

RotatE

RotatE defines each relation r as a rotation from an entity h to an entity t in the complex vector space.

Euler's formula: $e^{i\theta} = cos\theta + i\ sin\theta$, i.e., a unitary complex number tracing the unit circle in the complex plane as θ ranges through reals.

Encoder: Entities $h, t \in E$ and relations $r \in R$, through d-dimensional complex vectors $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{C}^d$, where each element r_i of $\mathbf{r}$ is of the form $e^{i\theta_{r,i}}$ (with modulus $| r_i | = 1$).

Decoder: Consider a dissimilarity measure $d(\mathbf{h} \odot \mathbf{r}, \mathbf{t}) = \|\mathbf{h} \odot \mathbf{r} - \mathbf{t}\|$, where $\odot$ denotes element-wise product, corresponding to a counterclockwise rotation in every dimension i by $\theta_{r,i}$ radians about the origin of the complex plane. Hence, $-\|\mathbf{h} \odot \mathbf{r} - \mathbf{t}\|$ defines a similarity measure.

RotatE

RotatE defines each relation r as a rotation from an entity h to an entity t in the complex vector space.

Euler's formula: $e^{i\theta} = cos\theta + i\ sin\theta$, i.e., a unitary complex number tracing the unit circle in the complex plane as θ ranges through reals.

Encoder: Entities $h, t \in E$ and relations $r \in R$, through d-dimensional complex vectors $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{C}^d$, where each element r_i of $\mathbf{r}$ is of the form $e^{i\theta_{r,i}}$ (with modulus $|\ r_i\ | = 1$).

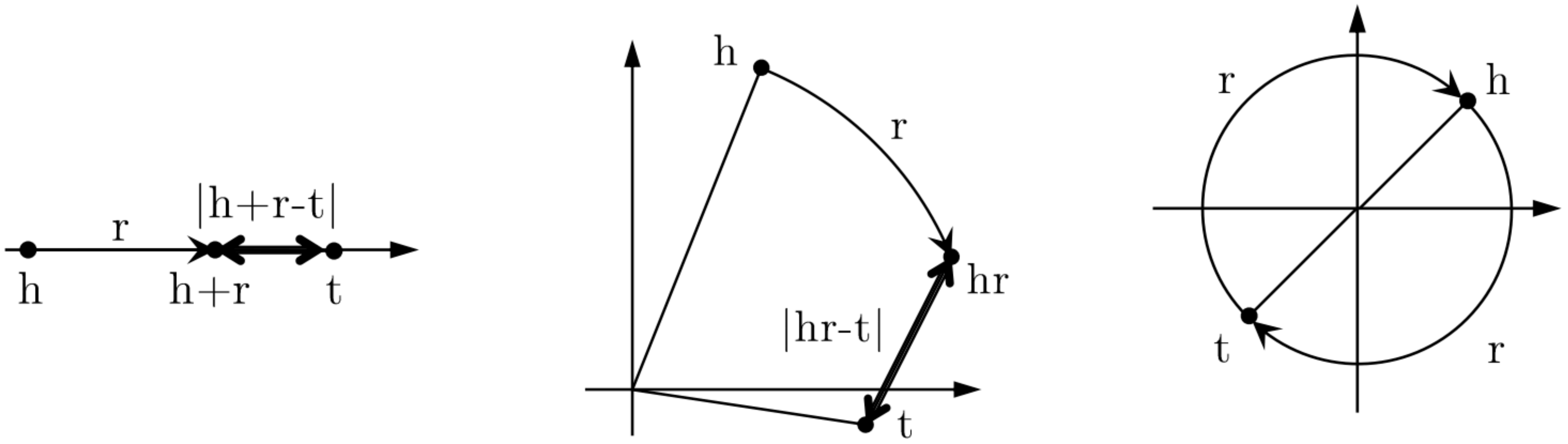
Decoder: Consider a dissimilarity measure $d(\mathbf{h} \odot \mathbf{r}, \mathbf{t}) = \|\mathbf{h} \odot \mathbf{r} - \mathbf{t}\|$, where $\odot$ denotes element-wise product, corresponding to a counterclockwise rotation in every dimension i by $\theta_{r,i}$ radians about the origin of the complex plane. Hence, $-\|\mathbf{h} \odot \mathbf{r} - \mathbf{t}\|$ defines a similarity measure.

Loss: For every fact $r(h, t)$, RotatE minimizes the following loss function:

$$\mathscr{L} = -\log \sigma(\gamma - d(\mathbf{h} \odot \mathbf{r}, \mathbf{t})) - \sum_{r(h',t') \in N^{r(h,t)}} \frac{1}{k} \log \sigma(d(\mathbf{h}' \odot \mathbf{r}, \mathbf{t}') - \gamma),$$

where γ is a fixed margin, σ is the sigmoid function, and $N^{r(h,t)}$ is a set of k negative samples for $r(h, t)$.

RotatE vs TransE



(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 (Sun et al): Comparing 1-dimensional embeddings of the models TransE and RotatE. Rotations in each individual dimension enable RotatE to capture symmetry.

RotatE can emulate TransE as a special case, see Theorem 4 of (Sun et al).

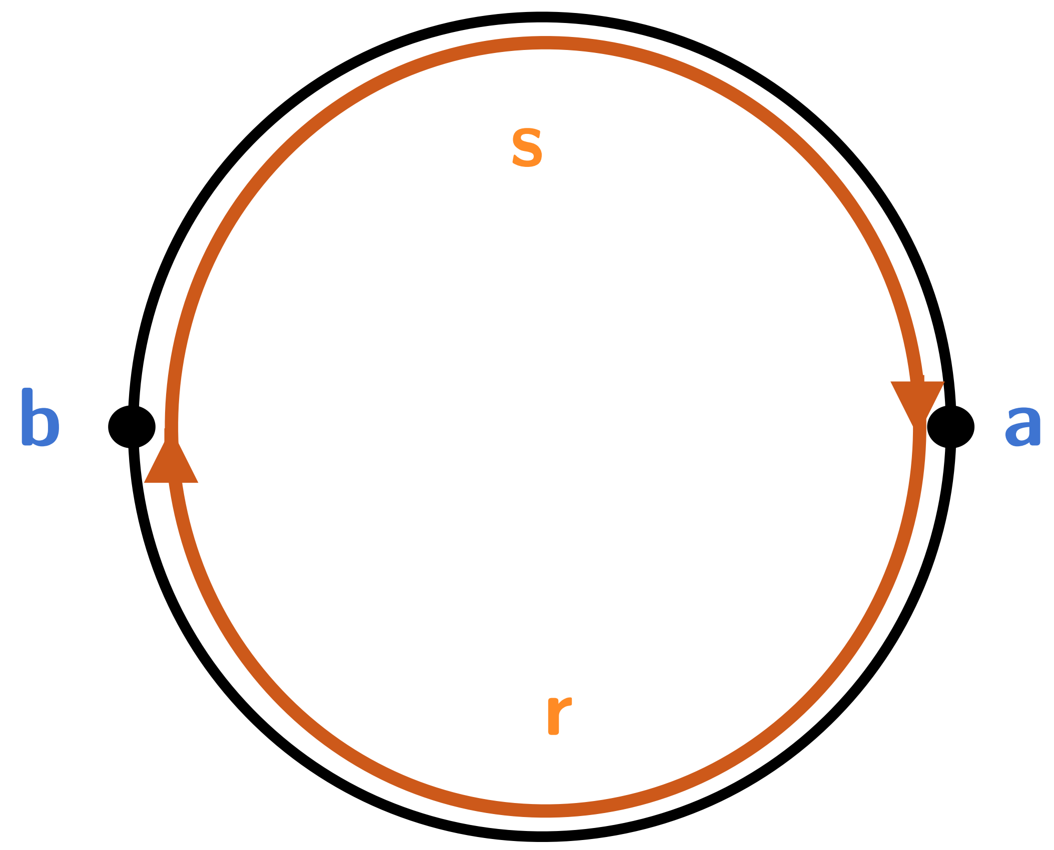
How Expressive is RotatE?

How Expressive is RotatE?

The facts $\{r(a,b), s(b,a)\}$ can be realized in RotatE by, e.g., the shown configuration.

How Expressive is RotatE?

The facts $\{r(a, b), s(b, a)\}$ can be realized in RotatE by, e.g., the shown configuration.



How Expressive is RotatE?

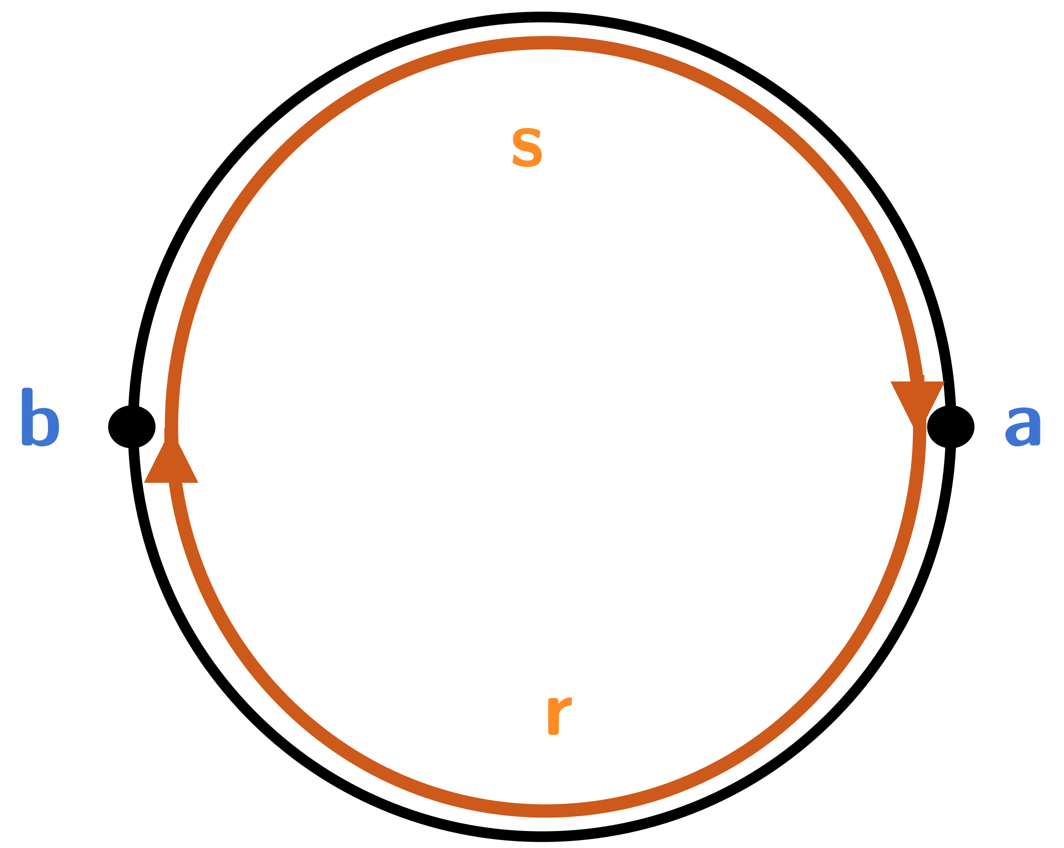
The facts $\{r(a,b), s(b,a)\}$ can be realized in RotatE by, e.g., the shown configuration.

To additionally realize $s(b,c)$, we need $\mathbf{a} \approx \mathbf{c}$ which implies, e.g., $r(c,b)$ as the rotation $\mathbf{r}$ from $\mathbf{c}$ results in $\mathbf{b}$.

RotatE sets $\mathbf{r}$ and $\mathbf{s}$ symmetric to capture the initial two facts, though the relations need not be symmetric.

RotatE cannot fit the sets facts:

$T = \{r(a,b), s(b,c), s(b,a)\}$ and $F = \{r(c,b)\}$.



How Expressive is RotatE?

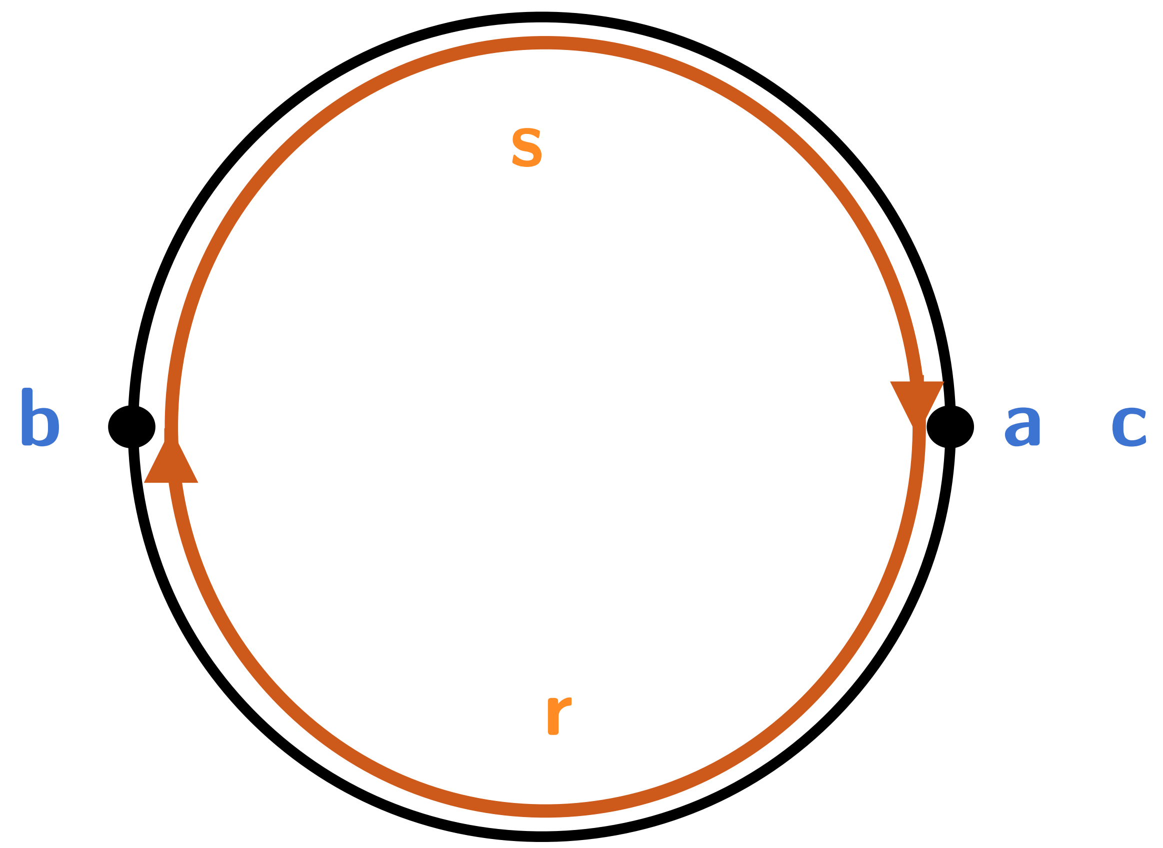
The facts $\{r(a,b), s(b,a)\}$ can be realized in RotatE by, e.g., the shown configuration.

To additionally realize $s(b,c)$, we need $\mathbf{a} \approx \mathbf{c}$ which implies, e.g., $r(c,b)$ as the rotation $\mathbf{r}$ from $\mathbf{c}$ results in $\mathbf{b}$.

RotatE sets $\mathbf{r}$ and $\mathbf{s}$ symmetric to capture the initial two facts, though the relations need not be symmetric.

RotatE cannot fit the sets facts:

$T = \{r(a,b), s(b,c), s(b,a)\}$ and $F = \{r(c,b)\}$.



Which Inference Patterns can RotatE capture?

Which Inference Patterns can RotatE capture?

All patterns captured by TransE can be captured by RotatE.

RotatE can also capture symmetry.

RotatE cannot capture the hierarchy pattern:
$\forall x, y \; r(x, y) \Rightarrow s(x, y)$.

To capture facts of the form $r(a, b), s(a, b), \ldots$ the rotations from a to b need to be similar, i.e., $\mathbf{r} \approx \mathbf{s}$, effectively enforcing relation equivalence.

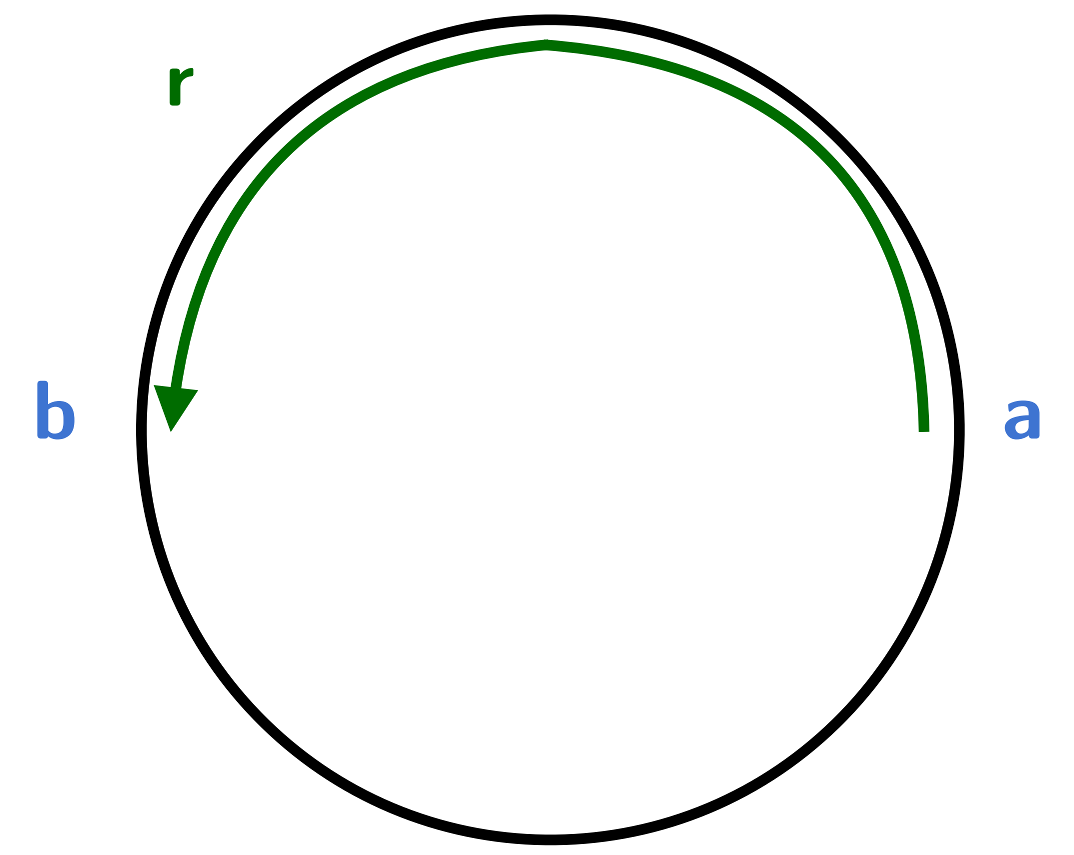
Which Inference Patterns can RotatE capture?

All patterns captured by TransE can be captured by RotatE.

RotatE can also capture symmetry.

RotatE cannot capture the hierarchy pattern:
$\forall x, y \; r(x, y) \Rightarrow s(x, y)$.

To capture facts of the form $r(a, b), s(a, b), \ldots$ the rotations from a to b need to be similar, i.e., $\mathbf{r} \approx \mathbf{s}$, effectively enforcing relation equivalence.



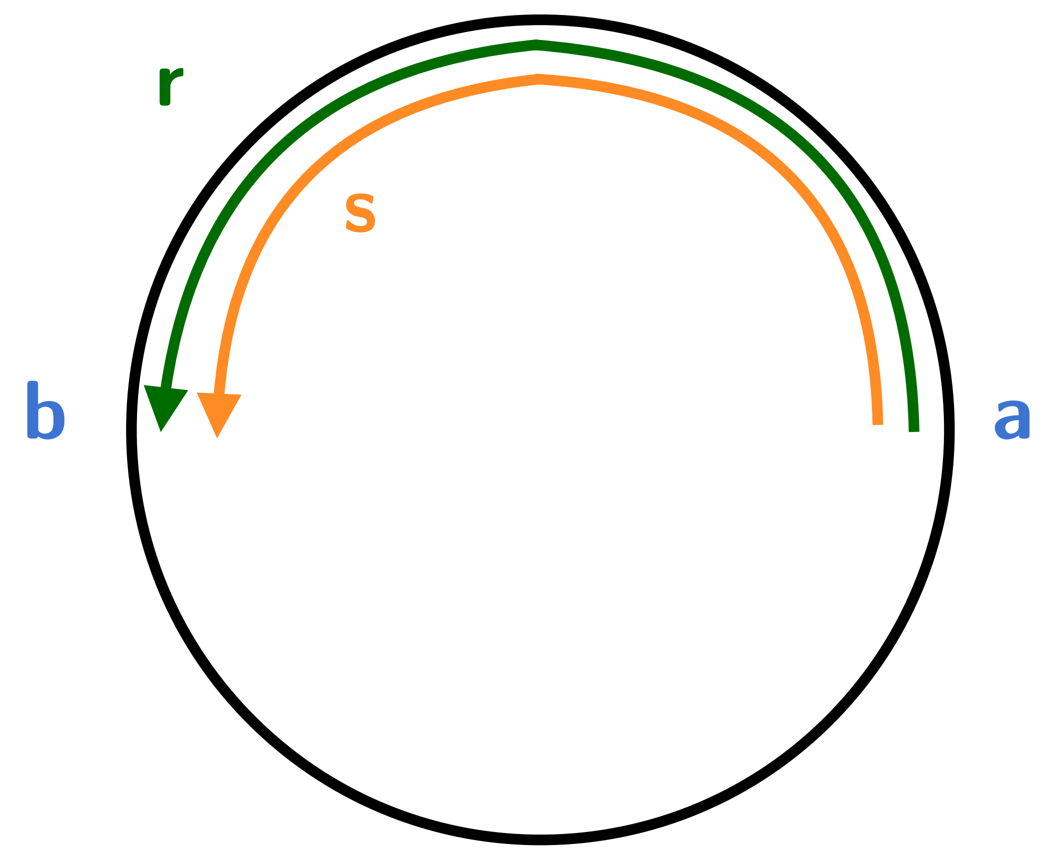
Which Inference Patterns can RotatE capture?

All patterns captured by TransE can be captured by RotatE.

RotatE can also capture symmetry.

RotatE cannot capture the hierarchy pattern:
$\forall x, y\ r(x, y) \Rightarrow s(x, y)$.

To capture facts of the form $r(a, b), s(a, b), \ldots$ the rotations from a to b need to be similar, i.e., $\mathbf{r} \approx \mathbf{s}$, effectively enforcing relation equivalence.



Bilinear Models

Tensor Representation of a KG

Tensor Representation of a KG

A KG G can be represented by defining, for every relation $r \in R$, an adjacency matrix $\mathbf{M}_r \in \mathbb{R}^{|E| \times |E|}$:

$$\mathbf{M}_{r[i,j]} = \begin{cases} 1 & \text{if } r(e_i, e_j) \in G, \\ 0 & \text{otherwise.} \end{cases}$$

Tensor Representation of a KG

A KG G can be represented by defining, for every relation $r \in R$, an adjacency matrix $\mathbf{M}_r \in \mathbb{R}^{|E| \times |E|}$:

$$\mathbf{M}_{r[i,j]} = \begin{cases} 1 & \text{if } r(e_i, e_j) \in G, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we can represent G in terms of a tensor $\mathbf{T} \in \mathbb{R}^{|E| \times |E| \times |R|}$:

$$\mathbf{T}_{i,j,k} = \begin{cases} 1 & \text{if } r_k(e_i, e_j) \in G, \\ 0 & \text{otherwise.} \end{cases}$$

Tensor Representation of a KG

A KG G can be represented by defining, for every relation $r \in R$, an adjacency matrix $\mathbf{M}_r \in \mathbb{R}^{|E| \times |E|}$:

$$\mathbf{M}_{r[i,j]} = \begin{cases} 1 & \text{if } r(e_i, e_j) \in G, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we can represent G in terms of a tensor $\mathbf{T} \in \mathbb{R}^{|E| \times |E| \times |R|}$:

$$\mathbf{T}_{i,j,k} = \begin{cases} 1 & \text{if } r_k(e_i, e_j) \in G, \\ 0 & \text{otherwise.} \end{cases}$$

Bilinear models use a bilinear product, to represent the relationships, hence the name "bilinear".

Bilinear models use tensor/matrix representation for relations and fall under tensor factorization methods.

RESCAL

$$\mathbf{h}^\top \begin{bmatrix} 0.5 & 1 & 0.2 \\ 1 & 0.2 & 0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}_r \mathbf{t}$$

RESCAL

$$\mathbf{h}^\top \begin{bmatrix} 0.5 & 1 & 0.2 \\ 1 & 0.2 & 0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}_r \mathbf{t}$$

RESCAL is the first bilinear model and has inspired a line of research.

Encoder: Entities $h, t \in E$ through vectors $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$, and relations $r \in R$, as a matrix $\mathbf{M}_r \in \mathbb{R}^d \times \mathbb{R}^d$.

Decoder: Scores a fact $r(h, t)$ as $\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$, which captures all interactions between the components of $\mathbf{h}$ and $\mathbf{t}$ and defines a similarity measure.

Loss: Exact formulation can vary, depending on regularization terms etc.

RESCAL

$$\mathbf{h}^\top \begin{bmatrix} 0.5 & 1 & 0.2 \\ 1 & 0.2 & 0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}_r \mathbf{t}$$

RESCAL

$$\mathbf{h}^\top \begin{bmatrix} 0.5 & 1 & 0.2 \\ 1 & 0.2 & 0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}_r \mathbf{t}$$

Expressiveness: RESCAL is fully expressive, as it is possible to fit arbitrary set of true and false facts using the power of full rank matrix. This requires $O(d^2)$ parameters per relation, and is impractical for large-scale KGs.

Problem: Using a full rank matrix is prone to overfitting, and this has motivated a line of research, where several restrictions are imposed on the representation.

DistMult

$$\mathbf{h}^\top \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}_r \mathbf{t}$$

DistMult

$$\mathbf{h}^\top \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}_r \mathbf{t}$$

DistMult is a bilinear model that restricts RESCAL to a diagonal matrix.

Encoder: DistMult does not allow an arbitrary matrix $\mathbf{M}_r \in \mathbb{R}^d \times \mathbb{R}^d$ for a relation $r \in R$ and restricts this to be the diagonal matrix $\mathbf{D}_r$.

Decoder: DistMult scores a fact $r(h, t)$ similar to RESCAL, with the restriction to the diagonal matrix: $\mathbf{h}^\top \mathbf{D}_r \mathbf{t}$.

DistMult

$$\mathbf{h}^\top \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}_r \mathbf{t}$$

DistMult

$$\mathbf{h}^\top \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}_r \mathbf{t}$$

Expressiveness: DistMult is not fully expressive since $\mathbf{h}^\top \mathbf{D}_r \mathbf{t} = \mathbf{t}^\top \mathbf{D}_r \mathbf{h}$.

DistMult cannot differentiate between head entity and tail entity: all relations are modeled as symmetric regardless, i.e., even anti-symmetric relations.

Scalability: While very inexpressive, DistMult is scalable, i.e., linear in d.

ComplEx

$$\mathbf{h}^\top \begin{bmatrix} 1+i & 0 & 0 \\ 0 & 2+i & 0 \\ 0 & 0 & 3+2i \end{bmatrix}_r \bar{\mathbf{t}}$$

ComplEx

$$\mathbf{h}^\top \begin{bmatrix} 1+i & 0 & 0 \\ 0 & 2+i & 0 \\ 0 & 0 & 3+2i \end{bmatrix}_r \bar{\mathbf{t}}$$

ComplEx is another bilinear model which extends DistMult to the complex domain.

Encoder: Entities $h, t \in E$ through d-dimensional vectors $\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$ in complex space, and relations $r \in R$, as a diagonal matrix $\mathbf{D}_r \in \mathbb{C}^d \times \mathbb{C}^d$ in this space.

Decoder: Scores a fact $r(h, t)$ as $\text{Re}(\mathbf{h}^\top \mathbf{D}_r \bar{\mathbf{t}})$, where $\bar{\mathbf{t}}$ defines the complex conjugate of $\mathbf{t}$, and Re denotes the real part of a complex vector.

ComplEx

$$\mathbf{h}^\top \begin{bmatrix} 1+i & 0 & 0 \\ 0 & 2+i & 0 \\ 0 & 0 & 3+2i \end{bmatrix}_r \bar{\mathbf{t}}$$

ComplEx

$$\mathbf{h}^\top \begin{bmatrix} 1+i & 0 & 0 \\ 0 & 2+i & 0 \\ 0 & 0 & 3+2i \end{bmatrix}_r \bar{\mathbf{t}}$$

Expressiveness: By the use of complex conjugates, ComplEx introduces asymmetry and thus can also model asymmetric relations. ComplEx is fully expressive for KGs.

ComplEx is an interesting trade-off, as it generalizes DistMult to a fully expressive model, while still using diagonal matrices, which are less prone to overfitting.

What Inference Patterns can Bilinear Models capture?

What Inference Patterns can Bilinear Models capture?

DistMult is inherently symmetric, no support for anti-symmetry.

What Inference Patterns can Bilinear Models capture?

DistMult is inherently symmetric, no support for anti-symmetry.

ComplEx can capture symmetry, anti-symmetry and inversion with the help of complex conjugates.

What Inference Patterns can Bilinear Models capture?

DistMult is inherently symmetric, no support for anti-symmetry.

ComplEx can capture symmetry, anti-symmetry and inversion with the help of complex conjugates.

Neither can capture composition (or intersection): The scoring functions of ComplEx or DistMult are not bijective, which is a necessary condition for capturing composition (Sun et al, 2019).

What Inference Patterns can Bilinear Models capture?

DistMult is inherently symmetric, no support for anti-symmetry.

ComplEx can capture symmetry, anti-symmetry and inversion with the help of complex conjugates.

Neither can capture composition (or intersection): The scoring functions of ComplEx or DistMult are not bijective, which is a necessary condition for capturing composition (Sun et al, 2019).

Both ComplEx and DistMult can capture the hierarchy pattern: For DistMult, set $s = \lambda r$, for a scalar $\lambda > 1$: Then $\mathbf{h}^\top \mathbf{D_r} \mathbf{t} < \mathbf{h}^\top \mathbf{D_s} \mathbf{t}$, and and hence $\forall x, y \; r(x, y) \Rightarrow s(x, y)$. The argument for ComplEx is analogous.

What Inference Patterns can Bilinear Models capture?

DistMult is inherently symmetric, no support for anti-symmetry.

ComplEx can capture symmetry, anti-symmetry and inversion with the help of complex conjugates.

Neither can capture composition (or intersection): The scoring functions of ComplEx or DistMult are not bijective, which is a necessary condition for capturing composition (Sun et al, 2019).

Both ComplEx and DistMult can capture the hierarchy pattern: For DistMult, set $s = \lambda r$, for a scalar $\lambda > 1$: Then $\mathbf{h}^\top \mathbf{D_r} \mathbf{t} < \mathbf{h}^\top \mathbf{D_s} \mathbf{t}$, and and hence $\forall x, y \; r(x, y) \Rightarrow s(x, y)$. The argument for ComplEx is analogous.

This does not mean that bilinear models can capture relational hierarchies: Hierarchies captured in bilinear models are inherently linear, and this is an important limitation.

What Inference Patterns can Bilinear Models capture?

DistMult is inherently symmetric, no support for anti-symmetry.

ComplEx can capture symmetry, anti-symmetry and inversion with the help of complex conjugates.

Neither can capture composition (or intersection): The scoring functions of ComplEx or DistMult are not bijective, which is a necessary condition for capturing composition (Sun et al, 2019).

Both ComplEx and DistMult can capture the hierarchy pattern: For DistMult, set $s = \lambda r$, for a scalar $\lambda > 1$: Then $\mathbf{h}^\top \mathbf{D_r t} < \mathbf{h}^\top \mathbf{D_s t}$, and and hence $\forall x, y\ r(x, y) \Rightarrow s(x, y)$. The argument for ComplEx is analogous.

This does not mean that bilinear models can capture relational hierarchies: Hierarchies captured in bilinear models are inherently linear, and this is an important limitation.

Models such as RESCAL and TuckER are same as ComplEx in terms of inference patterns.

Box Embedding Models

Box Embeddings

Box Embeddings

Box embedding models are translation-based approaches that make use of spatial features.

Box Embeddings

Box embedding models are translation-based approaches that make use of spatial features.

Entity classification: First used for entity classification, i.e., inferring the class of a given entity.

Box Embeddings

Box embedding models are translation-based approaches that make use of spatial features.

Entity classification: First used for entity classification, i.e., inferring the class of a given entity.

A probabilistic embedding model is proposed based on Box Lattice Measures (Vilnis et al.): every class (i.e., unary relation) and entity in a KG are represented by a box. Entity-class membership, as well as relation similarity, is captured by means of box intersection in the lattice representation space.

Box Embeddings

Box embedding models are translation-based approaches that make use of spatial features.

Entity classification: First used for entity classification, i.e., inferring the class of a given entity.

A probabilistic embedding model is proposed based on Box Lattice Measures (Vilnis et al.): every class (i.e., unary relation) and entity in a KG are represented by a box. Entity-class membership, as well as relation similarity, is captured by means of box intersection in the lattice representation space.

Oxford being a City is captured by two boxes, one for the Oxford entity and another for the city class, such that the Oxford box fits entirely into the City box.

Box Embeddings

Box embedding models are translation-based approaches that make use of spatial features.

Entity classification: First used for entity classification, i.e., inferring the class of a given entity.

A probabilistic embedding model is proposed based on Box Lattice Measures (Vilnis et al.): every class (i.e., unary relation) and entity in a KG are represented by a box. Entity-class membership, as well as relation similarity, is captured by means of box intersection in the lattice representation space.

Oxford being a City is captured by two boxes, one for the Oxford entity and another for the city class, such that the Oxford box fits entirely into the City box.

Query answering: Box embeddings are also used for query answering, i.e., answering queries over incomplete KGs: In Query2Box (Ren et al.), every query is represented in the embedding space, and operations, such as projection and intersection, are defined with the help of box embeddings.

Box Embeddings

Box embedding models are translation-based approaches that make use of spatial features.

Entity classification: First used for entity classification, i.e., inferring the class of a given entity.

A probabilistic embedding model is proposed based on Box Lattice Measures (Vilnis et al.): every class (i.e., unary relation) and entity in a KG are represented by a box. Entity-class membership, as well as relation similarity, is captured by means of box intersection in the lattice representation space.

Oxford being a City is captured by two boxes, one for the Oxford entity and another for the city class, such that the Oxford box fits entirely into the City box.

Query answering: Box embeddings are also used for query answering, i.e., answering queries over incomplete KGs: In Query2Box (Ren et al.), every query is represented in the embedding space, and operations, such as projection and intersection, are defined with the help of box embeddings.

Neither of these approaches facilitate means for using box embeddings for KG completion.

BoxE: Model Representation

BoxE: Model Representation

BoxE is a KG embedding model which uses boxes to represent relations.

BoxE: Model Representation

BoxE is a KG embedding model which uses boxes to represent relations.

BoxE applies to knowledge bases with higher-arity facts, but we focus on KGs, for ease of presentation.

BoxE: Model Representation

BoxE is a KG embedding model which uses boxes to represent relations.

BoxE applies to knowledge bases with higher-arity facts, but we focus on KGs, for ease of presentation.

Encoder: Each entity $h, t \in E$ in terms of two d-dimensional vectors $\mathbf{h}, \mathbf{b_h} \in \mathbb{R}^d$ and $\mathbf{t}, \mathbf{b_t} \in \mathbb{R}^d$; each (binary) relation $r \in R$, in terms of two d-dimensional hyper-rectangles, or boxes, $\mathbf{r^h}, \mathbf{r^t} \in \mathbb{R}^d$, corresponding to a head box and a tail box, respectively.

BoxE: Model Representation

BoxE is a KG embedding model which uses boxes to represent relations.

BoxE applies to knowledge bases with higher-arity facts, but we focus on KGs, for ease of presentation.

Encoder: Each entity $h, t \in E$ in terms of two d-dimensional vectors $\mathbf{h}, \mathbf{b_h} \in \mathbb{R}^d$ and $\mathbf{t}, \mathbf{b_t} \in \mathbb{R}^d$; each (binary) relation $r \in R$, in terms of two d-dimensional hyper-rectangles, or boxes, $\mathbf{r^h}, \mathbf{r^t} \in \mathbb{R}^d$, corresponding to a head box and a tail box, respectively.

Idea: The embedding $\mathbf{h}$ (resp., $\mathbf{t}$) defines the base position of an entity h (resp., t), and the embedding $\mathbf{b_h}$ (resp., $\mathbf{b_t}$) defines its translational bump, which translates other entities from their base positions to their final embeddings by "bumping" them.

BoxE: Model Representation

BoxE is a KG embedding model which uses boxes to represent relations.

BoxE applies to knowledge bases with higher-arity facts, but we focus on KGs, for ease of presentation.

Encoder: Each entity $h, t \in E$ in terms of two d-dimensional vectors $\mathbf{h}, \mathbf{b_h} \in \mathbb{R}^d$ and $\mathbf{t}, \mathbf{b_t} \in \mathbb{R}^d$; each (binary) relation $r \in R$, in terms of two d-dimensional hyper-rectangles, or boxes, $\mathbf{r^h}, \mathbf{r^t} \in \mathbb{R}^d$, corresponding to a head box and a tail box, respectively.

Idea: The embedding $\mathbf{h}$ (resp., $\mathbf{t}$) defines the base position of an entity h (resp., t), and the embedding $\mathbf{b_h}$ (resp., $\mathbf{b_t}$) defines its translational bump, which translates other entities from their base positions to their final embeddings by "bumping" them.

The final embedding of a head entity h relative to a fact $r(h, t)$ is given by: $\mathbf{h^{r(h,t)}} = \mathbf{h} + \mathbf{b_t}$.

BoxE: Model Representation

BoxE is a KG embedding model which uses boxes to represent relations.

BoxE applies to knowledge bases with higher-arity facts, but we focus on KGs, for ease of presentation.

Encoder: Each entity $h, t \in E$ in terms of two d-dimensional vectors $\mathbf{h}, \mathbf{b_h} \in \mathbb{R}^d$ and $\mathbf{t}, \mathbf{b_t} \in \mathbb{R}^d$; each (binary) relation $r \in R$, in terms of two d-dimensional hyper-rectangles, or boxes, $\mathbf{r^h}, \mathbf{r^t} \in \mathbb{R}^d$, corresponding to a head box and a tail box, respectively.

Idea: The embedding $\mathbf{h}$ (resp., $\mathbf{t}$) defines the base position of an entity h (resp., t), and the embedding $\mathbf{b_h}$ (resp., $\mathbf{b_t}$) defines its translational bump, which translates other entities from their base positions to their final embeddings by "bumping" them.

The final embedding of a head entity h relative to a fact $r(h, t)$ is given by: $\mathbf{h^{r(h,t)}} = \mathbf{h} + \mathbf{b_t}$.

The final embedding of a tail entity t relative to a fact $r(h, t)$ is given by: $\mathbf{t^{r(h,t)}} = \mathbf{t} + \mathbf{b_h}$.

BoxE: Scoring and Spatial Properties

BoxE: Scoring and Spatial Properties

Decoder: Consider a distance measure $\text{dist}(\mathbf{e}, \mathbf{B})$ which defines how close an entity embedding $\mathbf{e}$ is to the center of a box $\mathbf{B}$. BoxE scores a fact $r(h, t)$ as the sum of the L-x norms of such function:

$$\left\| \text{dist}(\mathbf{h}^{\mathbf{r}(\mathbf{h},\mathbf{t})}, \mathbf{r}^{\mathbf{h}}) \right\|_x + \left\| \text{dist}(\mathbf{t}^{\mathbf{r}(\mathbf{h},\mathbf{t})}, \mathbf{r}^{\mathbf{t}}) \right\|_x$$

As in other translational models, we can negate the term to frame it as a similarity measure.

BoxE: Scoring and Spatial Properties

Decoder: Consider a distance measure dist$(\mathbf{e}, \mathbf{B})$ which defines how close an entity embedding $\mathbf{e}$ is to the center of a box $\mathbf{B}$. BoxE scores a fact $r(h, t)$ as the sum of the L-x norms of such function:

$$\left\| \text{dist}(\mathbf{h}^{\mathbf{r}(\mathbf{h},\mathbf{t})}, \mathbf{r}^{\mathbf{h}}) \right\|_x + \left\| \text{dist}(\mathbf{t}^{\mathbf{r}(\mathbf{h},\mathbf{t})}, \mathbf{r}^{\mathbf{t}}) \right\|_x$$

As in other translational models, we can negate the term to frame it as a similarity measure.

Box sizes are dynamic and their position matters: Every relation may be represented with boxes of different size and their relative position in relation to entities are part of scoring.

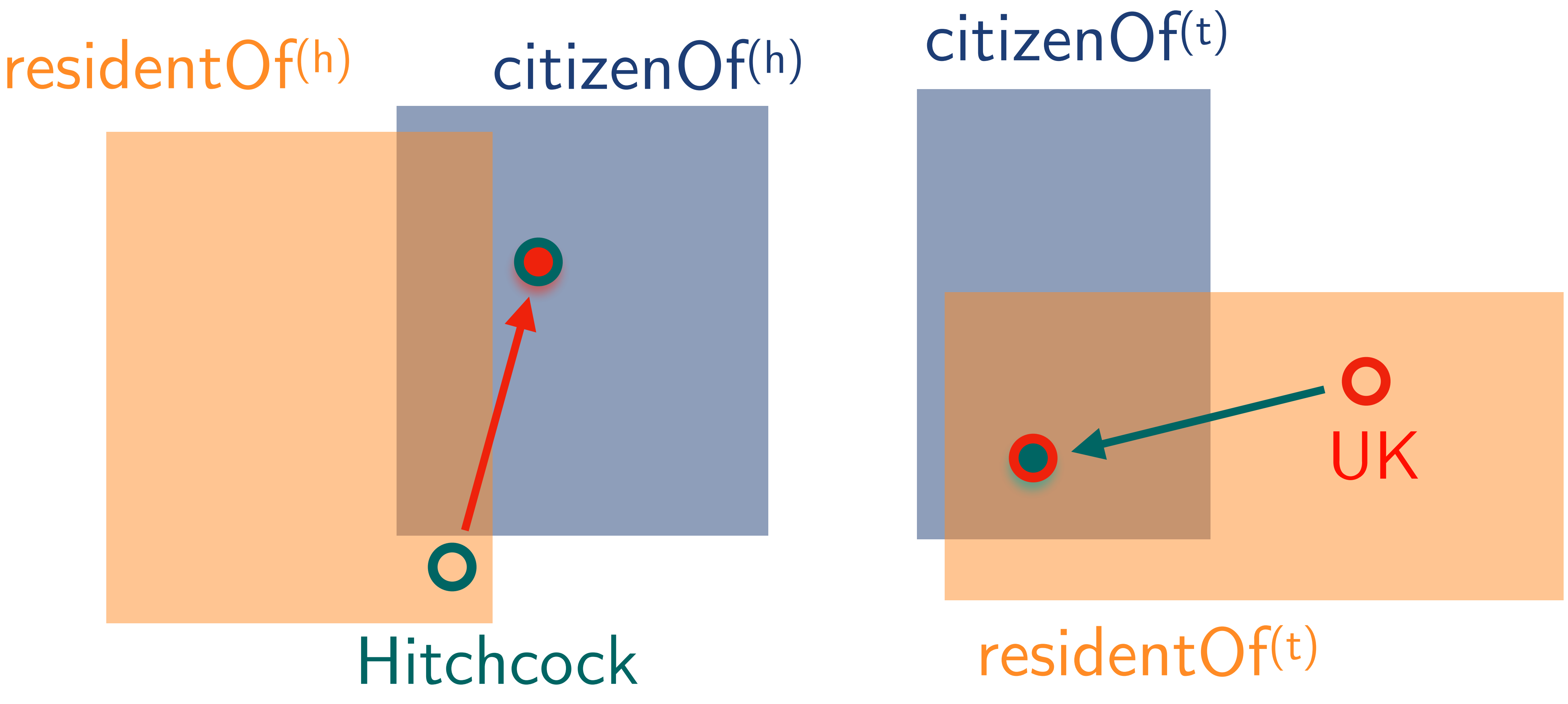
The final entity representation is dynamic: Every entity can have a potentially different final embedding relative to a different fact, since the bump vector depends on the other entity occurring in the fact.

How Expressive is BoxE?

How Expressive is BoxE?

A fact citizenOf(Hitchcock, UK) holds when the final embedding of the entity Hitchcock appears in the box $\text{citizenOf}^{(h)}$ and the the final embedding of the entity UK appears in the box $\text{citizenOf}^{(t)}$.

How Expressive is BoxE?

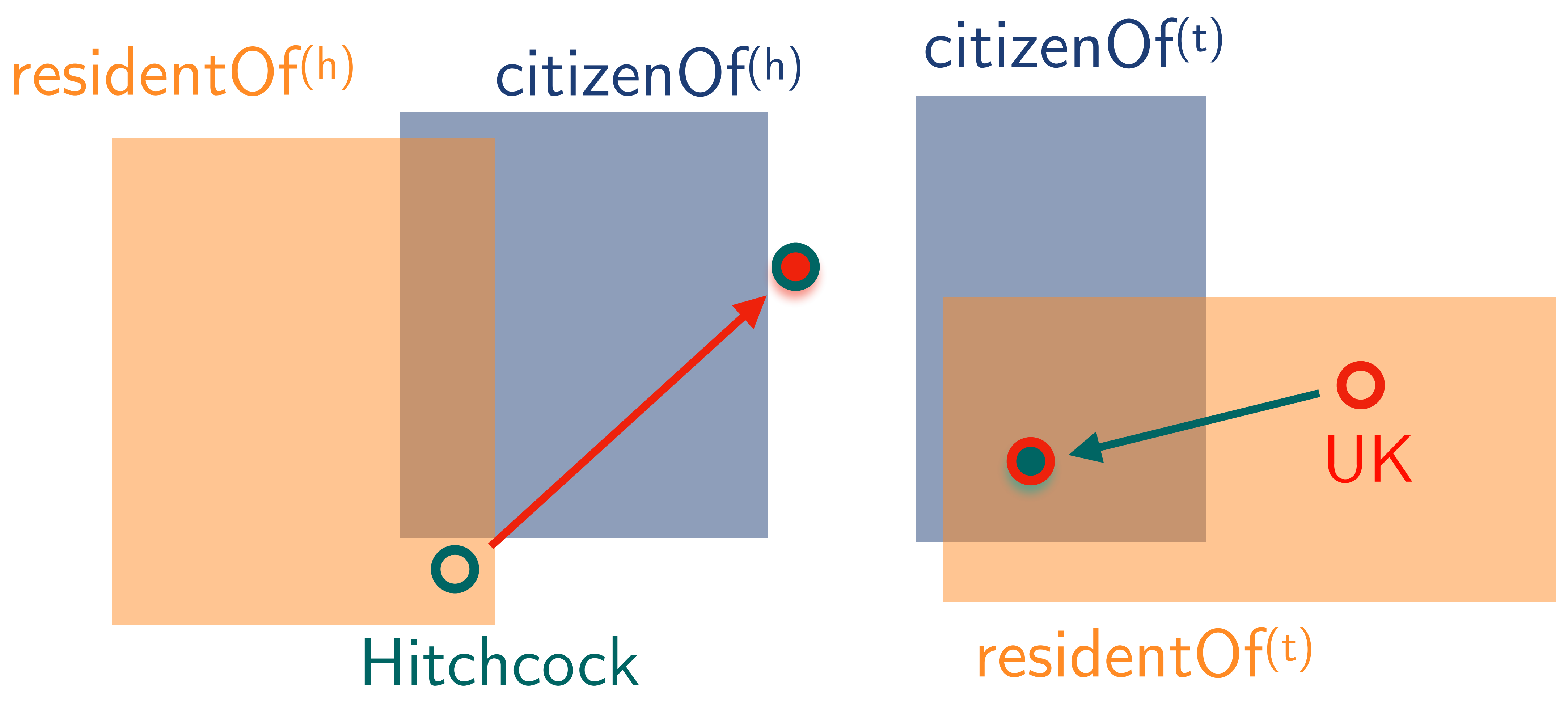


citizenOf(Hitchcock, UK) ✓

residentOf(Hitchcock, UK) ✗

A fact citizenOf(Hitchcock, UK) holds when the final embedding of the entity Hitchcock appears in the box citizenOf$^{(h)}$ and the the final embedding of the entity UK appears in the box citizenOf$^{(t)}$.

How Expressive is BoxE?

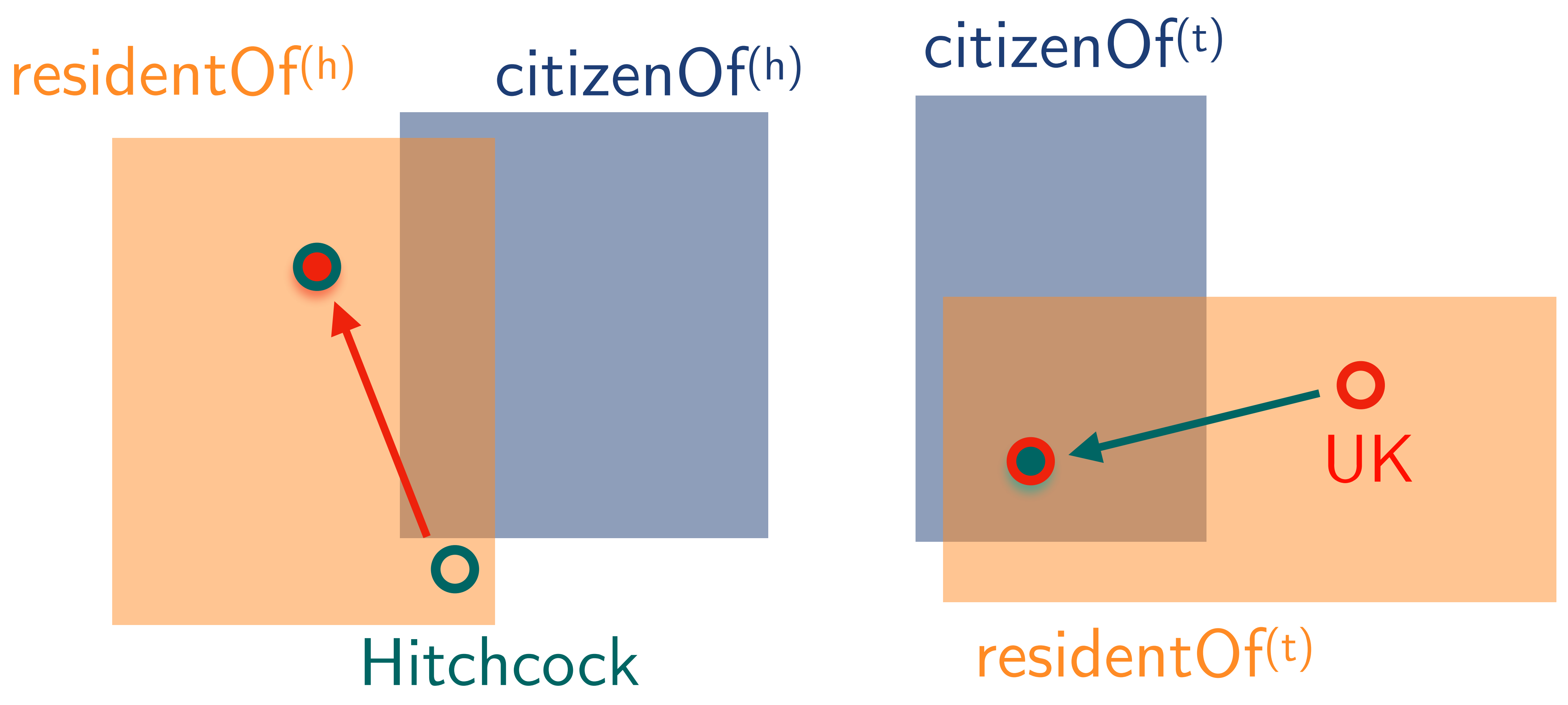


citizenOf(Hitchcock, UK) ✗

residentOf(Hitchcock, UK) ✗

A fact citizenOf(Hitchcock, UK) holds when the final embedding of the entity Hitchcock appears in the box citizenOf$^{(h)}$ and the the final embedding of the entity UK appears in the box citizenOf$^{(t)}$.

How Expressive is BoxE?

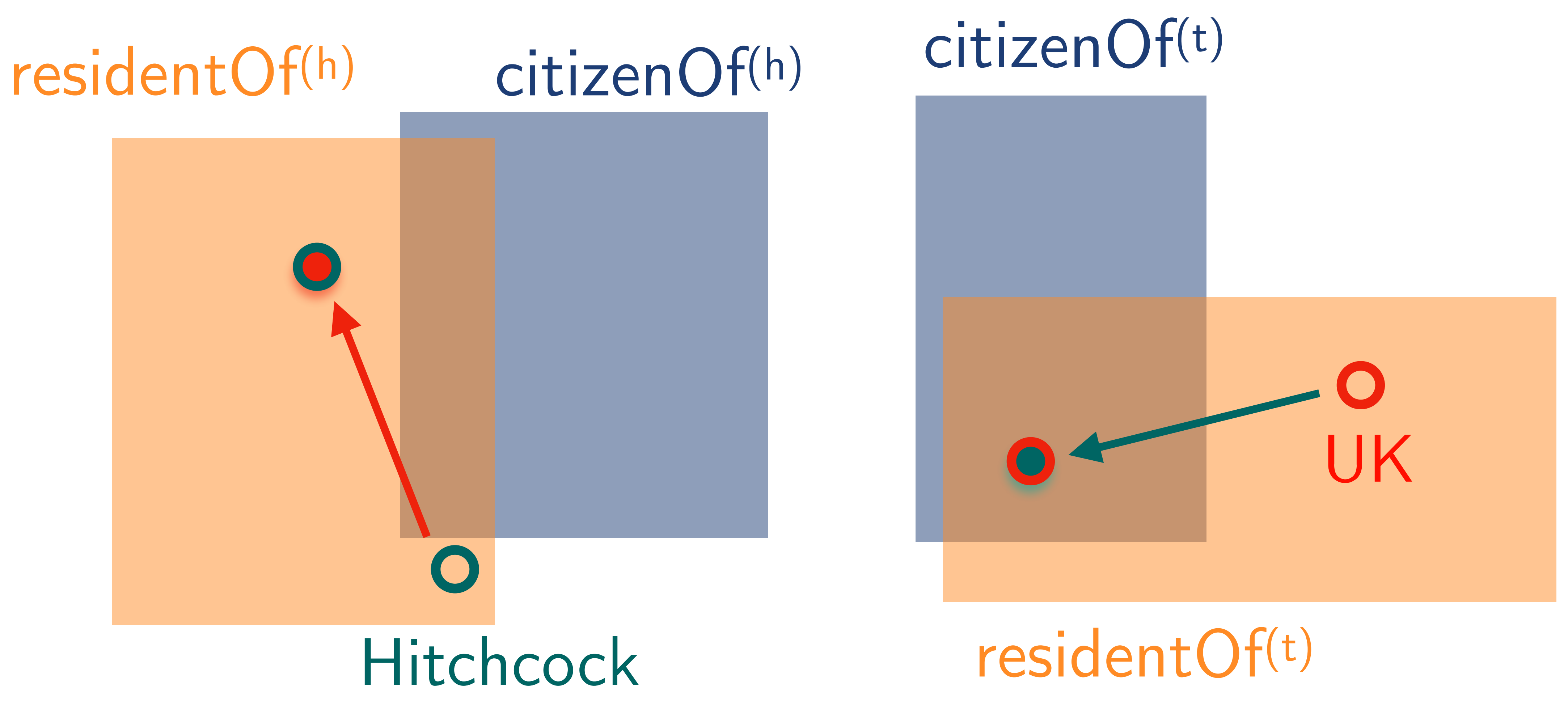


citizenOf(Hitchcock, UK) ✗

residentOf(Hitchcock, UK) ✓

A fact citizenOf(Hitchcock, UK) holds when the final embedding of the entity Hitchcock appears in the box citizenOf$^{(h)}$ and the the final embedding of the entity UK appears in the box citizenOf$^{(t)}$.

How Expressive is BoxE?



citizenOf(Hitchcock, UK) ✗

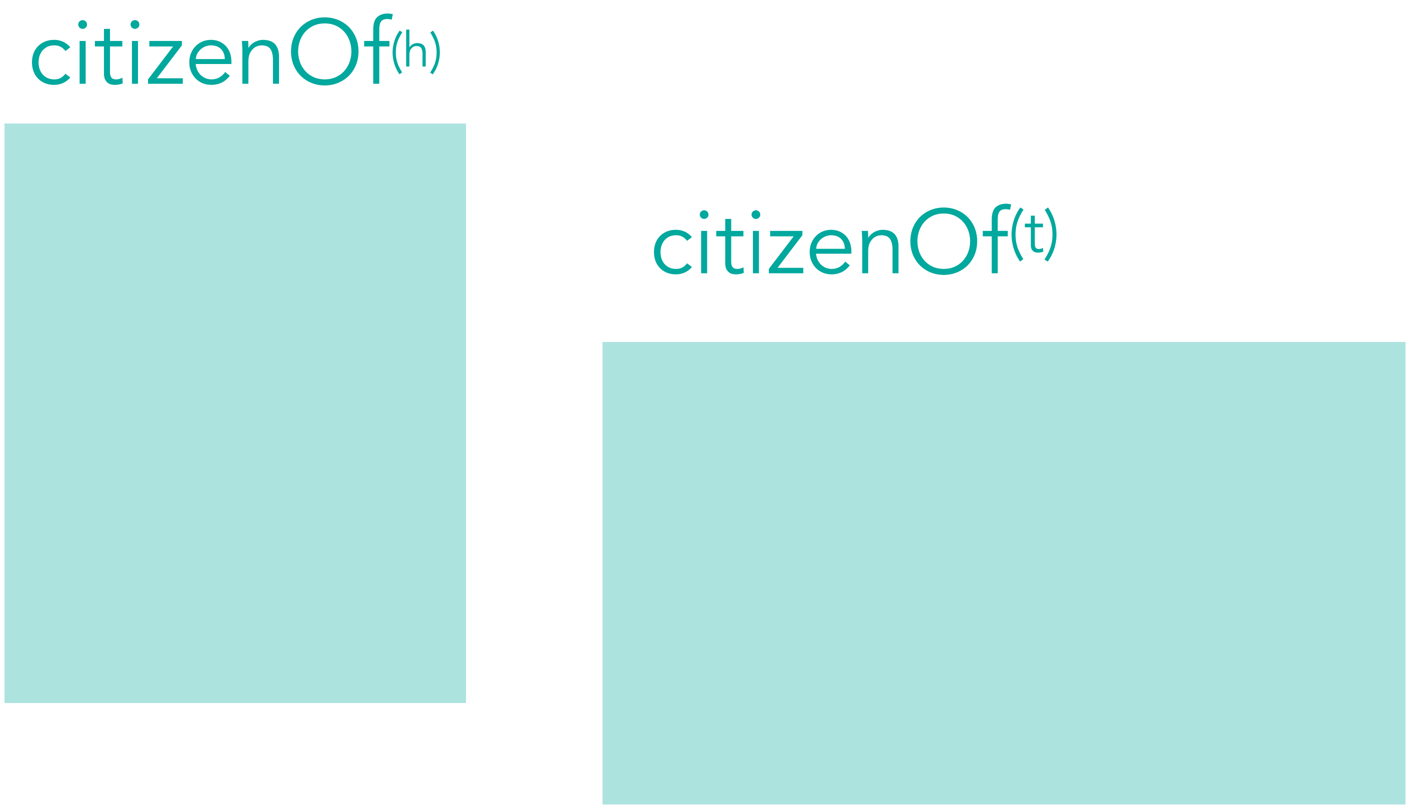
residentOf(Hitchcock, UK) ✓

A fact citizenOf(Hitchcock, UK) holds when the final embedding of the entity Hitchcock appears in the box citizenOf(h) and the the final embedding of the entity UK appears in the box citizenOf(t).

Expressiveness: BoxE is fully expressive. Any fact $r(h,t)$ can be made false in the model, by defining a bump vector for, e.g., the head entity h such that the tail entity t is pushed outside of the tail box of r in a single dimension. This operation can be done for all false facts without "harming" true facts, using $E \times R$ dimensions.

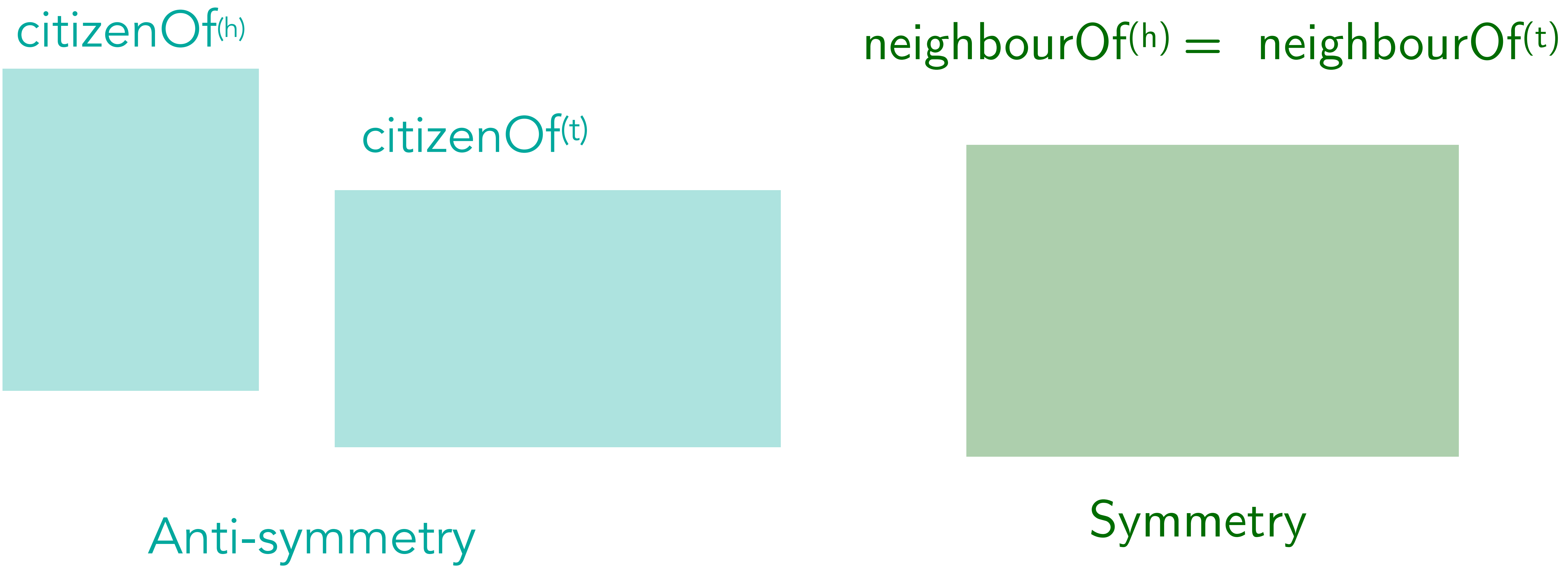
What Inference Patterns can BoxE capture?

What Inference Patterns can BoxE capture?

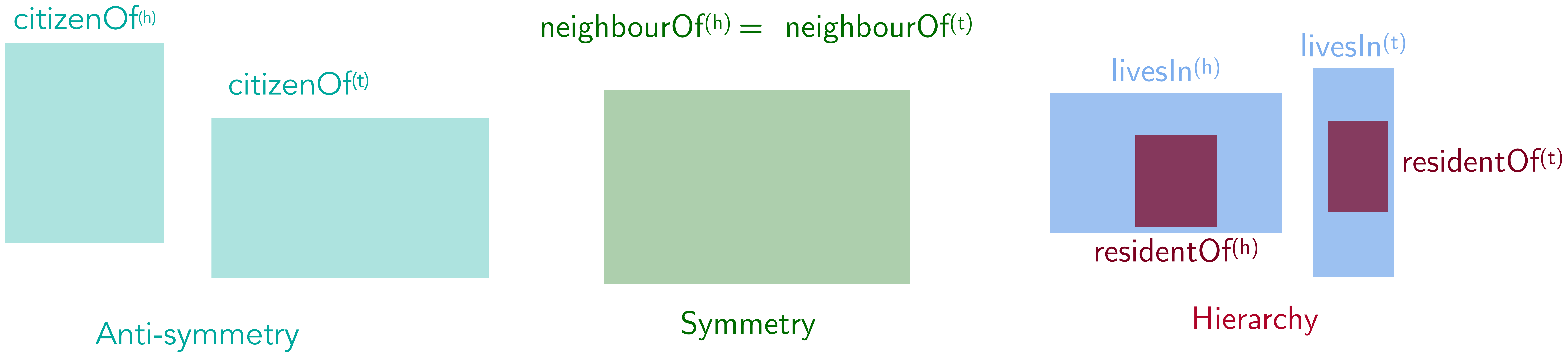


Anti-symmetry

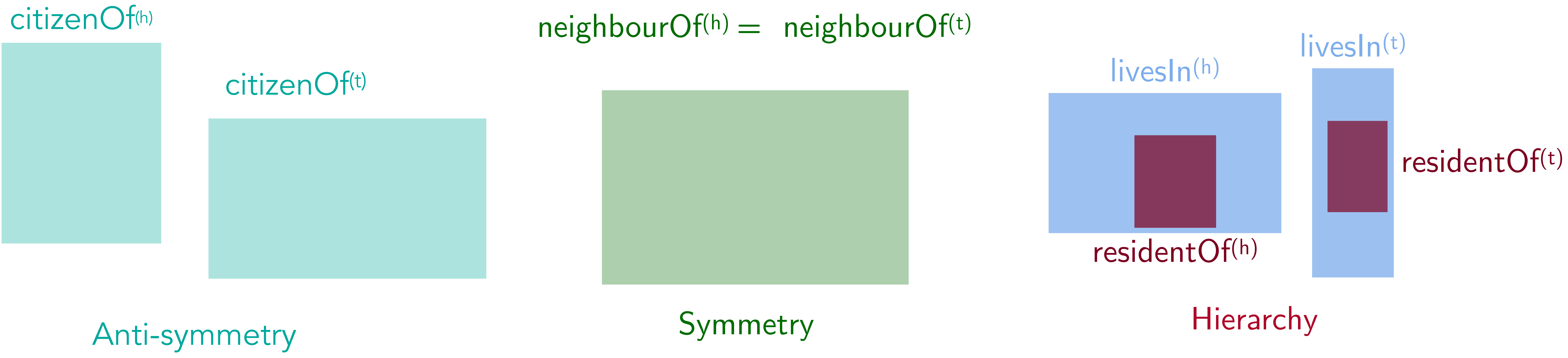
What Inference Patterns can BoxE capture?



What Inference Patterns can BoxE capture?

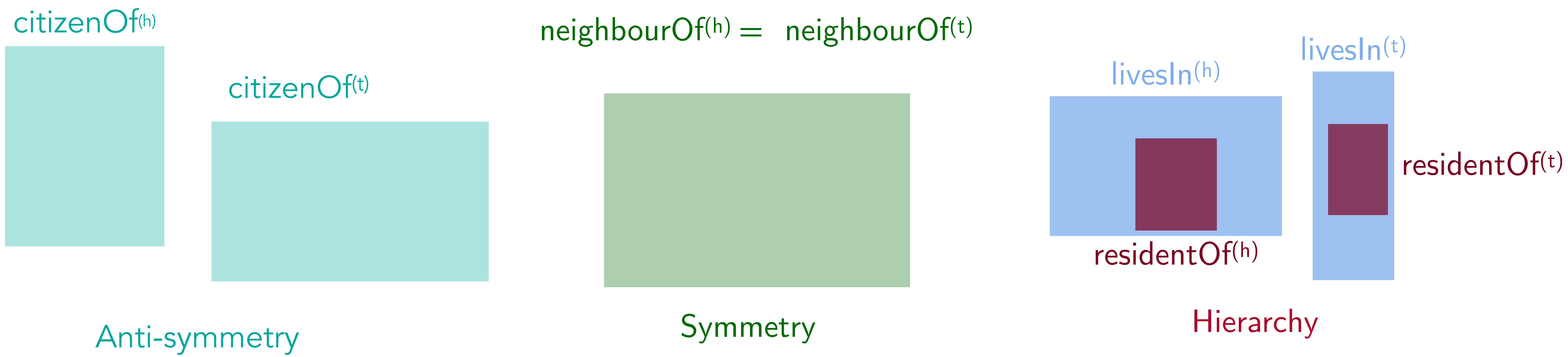


What Inference Patterns can BoxE capture?



Other inference patterns, e.g., inverse, mutual exclusion, intersection can be captured by configuring boxes.

What Inference Patterns can BoxE capture?



Other inference patterns, e.g., inverse, mutual exclusion, intersection can be captured by configuring boxes.

This approach does not work for the composition pattern: $\forall x, y, z \ r(x, y) \wedge s(y, z) \Rightarrow t(x, z)$! BoxE cannot capture composition as an inference pattern.

Generalized Inference Patterns

Generalized Inference Patterns

Inference patterns only cover a single application of a rule (Abboud et al., 2020).

Question: Can a model capture multiple instances of the same inference pattern jointly?

Capturing generalized inference patterns turns out to be significantly more challenging...

Generalized Inference Patterns

Inference patterns only cover a single application of a rule (Abboud et al., 2020).

Question: Can a model capture multiple instances of the same inference pattern jointly?

Capturing generalized inference patterns turns out to be significantly more challenging...

Example: TransE or RotatE can separately capture the composition rules:

$$\forall x, y, z \; r_1(x, y) \wedge r_4(y, z) \Rightarrow r_3(x, z) \text{ and } \forall x, y, z \; r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z),$$

but jointly capturing these incorrectly forces relation equivalence between r_2 and r_4.

Generalized Inference Patterns

Inference patterns only cover a single application of a rule (Abboud et al., 2020).

Question: Can a model capture multiple instances of the same inference pattern jointly?

Capturing generalized inference patterns turns out to be significantly more challenging...

Example: TransE or RotatE can separately capture the composition rules:

$$\forall x, y, z \ r_1(x, y) \wedge r_4(y, z) \Rightarrow r_3(x, z) \text{ and } \forall x, y, z \ r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z),$$

but jointly capturing these incorrectly forces relation equivalence between r_2 and r_4.

Example: Bilinear models can separately capture the hierarchy rules:

$$\forall x, y \ r_1(x, y) \Rightarrow r_3(x, y) \text{ and } \forall x, y \ r_2(x, y) \Rightarrow r_3(x, y).$$

Jointly capturing these imposes either $\forall x, y \ r_1(x, y) \Rightarrow r_2(x, y)$ or $\forall x, y \ r_2(x, y) \Rightarrow r_1(x, y)$ (Gutiérrez-Basulto et al.).

Generalized Inference Patterns

Inference patterns only cover a single application of a rule (Abboud et al., 2020).

Question: Can a model capture multiple instances of the same inference pattern jointly?

Capturing generalized inference patterns turns out to be significantly more challenging...

Example: TransE or RotatE can separately capture the composition rules:

$$\forall x, y, z \; r_1(x, y) \wedge r_4(y, z) \Rightarrow r_3(x, z) \text{ and } \forall x, y, z \; r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z),$$

but jointly capturing these incorrectly forces relation equivalence between r_2 and r_4.

Example: Bilinear models can separately capture the hierarchy rules:

$$\forall x, y \; r_1(x, y) \Rightarrow r_3(x, y) \text{ and } \forall x, y \; r_2(x, y) \Rightarrow r_3(x, y).$$

Jointly capturing these imposes either $\forall x, y \; r_1(x, y) \Rightarrow r_2(x, y)$ or $\forall x, y \; r_2(x, y) \Rightarrow r_1(x, y)$ (Gutiérrez-Basulto et al.).

A simple relational hierarchy cannot be captured by any of these systems. BoxE can capture these inference patterns also in this general sense, and can capture, e.g., relational hierarchies.

Rule Languages

Rule Languages

Even generalized inference patterns are limited: r_1, r_2 composes to r_3, and r_1, r_3 are symmetric (Abboud et al., 2020).

Question: Can a model capture different inference patterns jointly?

Rule languages: a simple rule language is a union of inference rules: symmetry, anti-symmetry, hierarchy, etc...

Rule Languages

Even generalized inference patterns are limited: r_1, r_2 composes to r_3, and r_1, r_3 are symmetric (Abboud et al., 2020).

Question: Can a model capture different inference patterns jointly?

Rule languages: a simple rule language is a union of inference rules: symmetry, anti-symmetry, hierarchy, etc...

Example: RotatE can separately capture each of the rules:

$$\forall x, y, z \; cousins(x, y) \wedge hasChild(y, z) \rightarrow relatives(x, z),$$

$$\forall x, y \; cousins(x, y) \rightarrow cousins(y, x),$$

$$\forall x, y \; relatives(x, y) \rightarrow relatives(y, x),$$

but jointly capturing these incorrectly forces $\forall x, y \; hasChild(x, y) \rightarrow hasChild(y, x)$!

Rule Languages

Even generalized inference patterns are limited: r_1, r_2 composes to r_3, and r_1, r_3 are symmetric (Abboud et al., 2020).

Question: Can a model capture different inference patterns jointly?

Rule languages: a simple rule language is a union of inference rules: symmetry, anti-symmetry, hierarchy, etc...

Example: RotatE can separately capture each of the rules:

$$\forall x, y, z \; cousins(x, y) \wedge hasChild(y, z) \rightarrow relatives(x, z),$$

$$\forall x, y \; cousins(x, y) \rightarrow cousins(y, x),$$

$$\forall x, y \; relatives(x, y) \rightarrow relatives(y, x),$$

but jointly capturing these incorrectly forces $\forall x, y \; hasChild(x, y) \rightarrow hasChild(y, x)$!

To better assess the inductive capacity of a model, show the rule language it can capture.

Overview of Embedding Models

Embedding Models: Representation and Scoring

Model	Entity representation	Relation representation	Scoring function
TransE	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$
RotatE	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$-\|\mathbf{h} \odot \mathbf{r} - \mathbf{t}\|$
RESCAL	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^d \times \mathbb{R}^d$	$\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$
DistMult	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{D}_r \in \mathbb{R}^d \times \mathbb{R}^d$	$\mathbf{h}^\top \mathbf{D}_r \mathbf{t}$
ComplEx	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{D}_r \in \mathbb{C}^d \times \mathbb{C}^d$	$\mathrm{Re}(\mathbf{h}^\top \mathbf{D}_r \bar{\mathbf{t}})$
BoxE	$\mathbf{h}, \mathbf{t}, \mathbf{b_h}, \mathbf{b_t} \in \mathbb{R}^d$	Hyper-rect's $\mathbf{r^h}, \mathbf{r^t} \in \mathbb{R}^d$	$-\left(\left\| \mathrm{dist}(\mathbf{h}^{\mathbf{r}(\mathbf{h},\mathbf{t})}, \mathbf{r^h}) \right\|_x + \left\| \mathrm{dist}(\mathbf{t}^{\mathbf{r}(\mathbf{h},\mathbf{t})}, \mathbf{r^t}) \right\|_x \right)$

Embedding Models: Representation and Scoring

Model	Entity representation	Relation representation	Scoring function
TransE	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\lVert \mathbf{h} + \mathbf{r} - \mathbf{t} \rVert$
RotatE	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$-\lVert \mathbf{h} \odot \mathbf{r} - \mathbf{t} \rVert$
RESCAL	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^d \times \mathbb{R}^d$	$\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$
DistMult	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{D}_r \in \mathbb{R}^d \times \mathbb{R}^d$	$\mathbf{h}^\top \mathbf{D}_r \mathbf{t}$
ComplEx	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{D}_r \in \mathbb{C}^d \times \mathbb{C}^d$	$\mathrm{Re}(\mathbf{h}^\top \mathbf{D}_r \bar{\mathbf{t}})$
BoxE	$\mathbf{h}, \mathbf{t}, \mathbf{b}_\mathbf{h}, \mathbf{b}_\mathbf{t} \in \mathbb{R}^d$	Hyper-rect's $\mathbf{r}^\mathbf{h}, \mathbf{r}^\mathbf{t} \in \mathbb{R}^d$	$-\left(\lVert \mathrm{dist}(\mathbf{h}^{\mathbf{r}(\mathbf{h},\mathbf{t})}, \mathbf{r}^\mathbf{h}) \rVert_x + \lVert \mathrm{dist}(\mathbf{t}^{\mathbf{r}(\mathbf{h},\mathbf{t})}, \mathbf{r}^\mathbf{t}) \rVert_x \right)$

Summary of the models covered in the lecture: Entity representations $h, t \in E$ and relation representations $r \in R$ are given, along with the scoring function for an arbitrary fact $r(h, t)$. Please refer to the original works for the details.

Embedding Models: Expressiveness and Inferences

Inference pattern	**TransE**	**RotatE**	**BoxE**	**DistMult**	**ComplEx**
Symmetry	N/N	Y/Y	Y/Y	Y/Y	Y/Y
Anti-symmetry	Y/Y	Y/Y	Y/Y	N/N	Y/Y
Inversion	Y/N	Y/Y	Y/Y	N/N	Y/Y
Composition	Y/N	Y/N	N/N	N/N	N/N
Hierarchy	N/N	N/N	Y/Y	Y/N	Y/N
Intersection	Y/N	Y/N	Y/Y	N/N	N/N
Mutual exclusion	Y/Y	Y/Y	Y/Y	Y/N	Y/N

Embedding Models: Expressiveness and Inferences

Inference pattern	TransE	RotatE	BoxE	DistMult	ComplEx
Symmetry	N/N	Y/Y	Y/Y	Y/Y	Y/Y
Anti-symmetry	Y/Y	Y/Y	Y/Y	N/N	Y/Y
Inversion	Y/N	Y/Y	Y/Y	N/N	Y/Y
Composition	Y/N	Y/N	N/N	N/N	N/N
Hierarchy	N/N	N/N	Y/Y	Y/N	Y/N
Intersection	Y/N	Y/N	Y/Y	N/N	N/N
Mutual exclusion	Y/Y	Y/Y	Y/Y	Y/N	Y/N

A summary of the inference patterns / generalized inference patterns that can be captured by selected models.

Embedding Models: Expressiveness and Inferences

Inference pattern	TransE	RotatE	BoxE	DistMult	ComplEx
Symmetry	N/N	Y/Y	Y/Y	Y/Y	Y/Y
Anti-symmetry	Y/Y	Y/Y	Y/Y	N/N	Y/Y
Inversion	Y/N	Y/Y	Y/Y	N/N	Y/Y
Composition	Y/N	Y/N	N/N	N/N	N/N
Hierarchy	N/N	N/N	Y/Y	Y/N	Y/N
Intersection	Y/N	Y/N	Y/Y	N/N	N/N
Mutual exclusion	Y/Y	Y/Y	Y/Y	Y/N	Y/N

A summary of the inference patterns / generalized inference patterns that can be captured by selected models.

Another bilinear model TuckER, coincides with ComplEX in terms of the listed inference patterns.

Neural Models

Neural Models

We have not discussed neural models and focused on so-called shallow embedding models so far.

Neural Models

We have not discussed neural models and focused on so-called shallow embedding models so far.

General approach: Neural models either use a neural network as a scoring function (e.g., ConvE), or use existing embedding models for scoring, but learn the embeddings with a neural network (e.g., r-GCN).

Neural Models

We have not discussed neural models and focused on so-called shallow embedding models so far.

General approach: Neural models either use a neural network as a scoring function (e.g., ConvE), or use existing embedding models for scoring, but learn the embeddings with a neural network (e.g., r-GCN).

Expressiveness vs Interpretability: Neural models are typically expressive, but they are hard to interpret and evaluate, since they are mostly black-box.

Neural Models

We have not discussed neural models and focused on so-called shallow embedding models so far.

General approach: Neural models either use a neural network as a scoring function (e.g., ConvE), or use existing embedding models for scoring, but learn the embeddings with a neural network (e.g., r-GCN).

Expressiveness vs Interpretability: Neural models are typically expressive, but they are hard to interpret and evaluate, since they are mostly black-box.

Practical: Shallow embedding models are state-of-the-art on many benchmark datasets.

Neural Models

We have not discussed neural models and focused on so-called shallow embedding models so far.

General approach: Neural models either use a neural network as a scoring function (e.g., ConvE), or use existing embedding models for scoring, but learn the embeddings with a neural network (e.g., r-GCN).

Expressiveness vs Interpretability: Neural models are typically expressive, but they are hard to interpret and evaluate, since they are mostly black-box.

Practical: Shallow embedding models are state-of-the-art on many benchmark datasets.

Conceptual: Shallow approaches are inherently transductive (i.e., limited to the entities they are trained on; see, e.g., (Hamilton et al., 2017)), while some neural models learn inductive representations (i.e., once learned, they can be applied to unseen entities).

Neural Models

We have not discussed neural models and focused on so-called shallow embedding models so far.

General approach: Neural models either use a neural network as a scoring function (e.g., ConvE), or use existing embedding models for scoring, but learn the embeddings with a neural network (e.g., r-GCN).

Expressiveness vs Interpretability: Neural models are typically expressive, but they are hard to interpret and evaluate, since they are mostly black-box.

Practical: Shallow embedding models are state-of-the-art on many benchmark datasets.

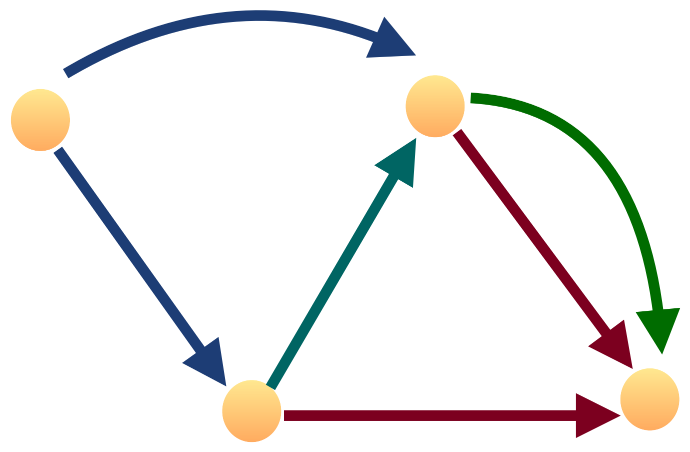
Conceptual: Shallow approaches are inherently transductive (i.e., limited to the entities they are trained on; see, e.g., (Hamilton et al., 2017)), while some neural models learn inductive representations (i.e., once learned, they can be applied to unseen entities).

We will revisit knowledge graph completion in the context of graph neural networks.

Temporal Knowledge Graph Completion

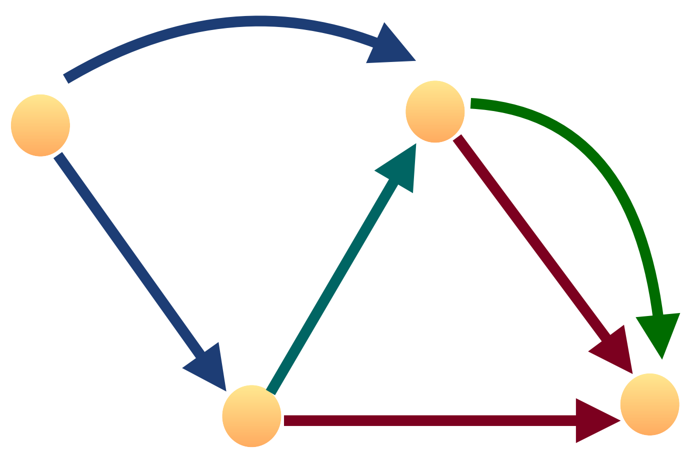
Temporal Knowledge Graphs

Temporal Knowledge Graphs

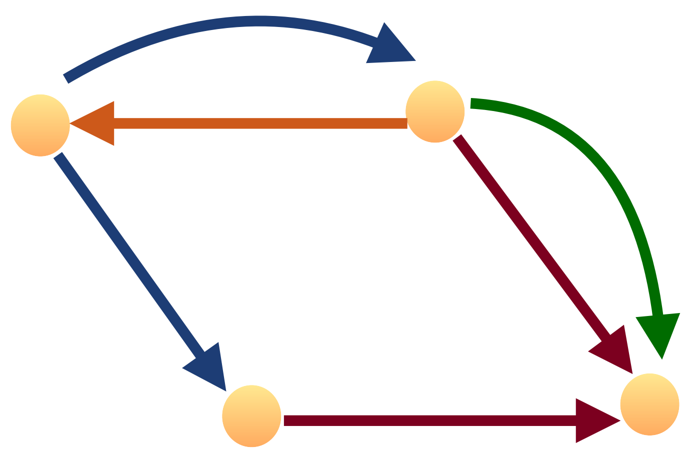


t_1

Temporal Knowledge Graphs

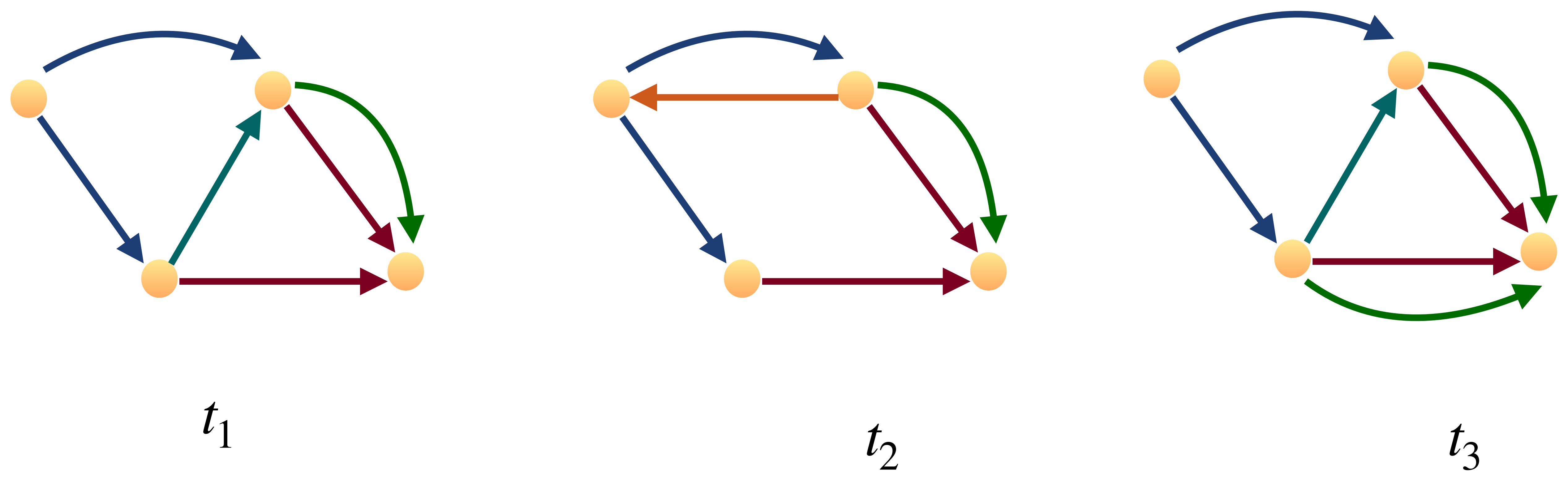


t_1

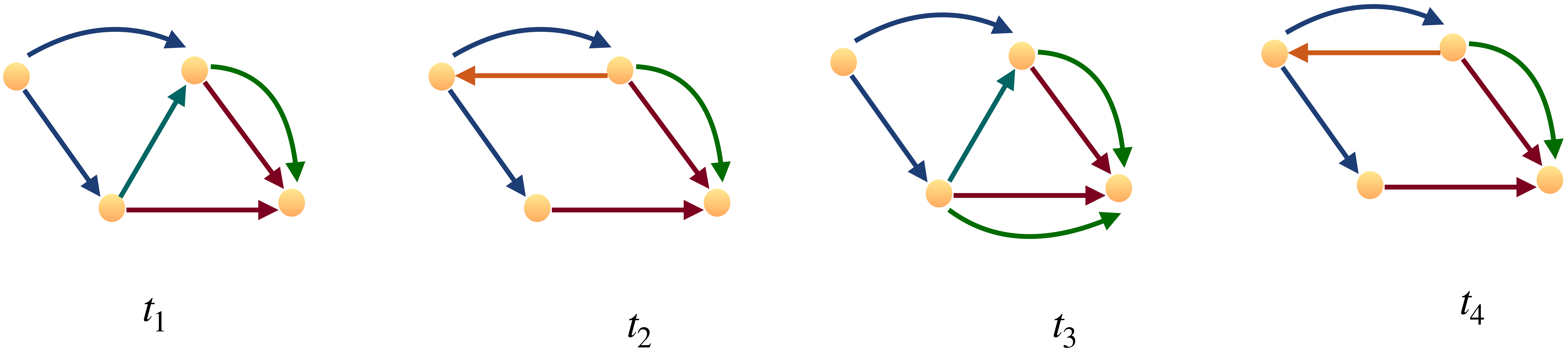


t_2

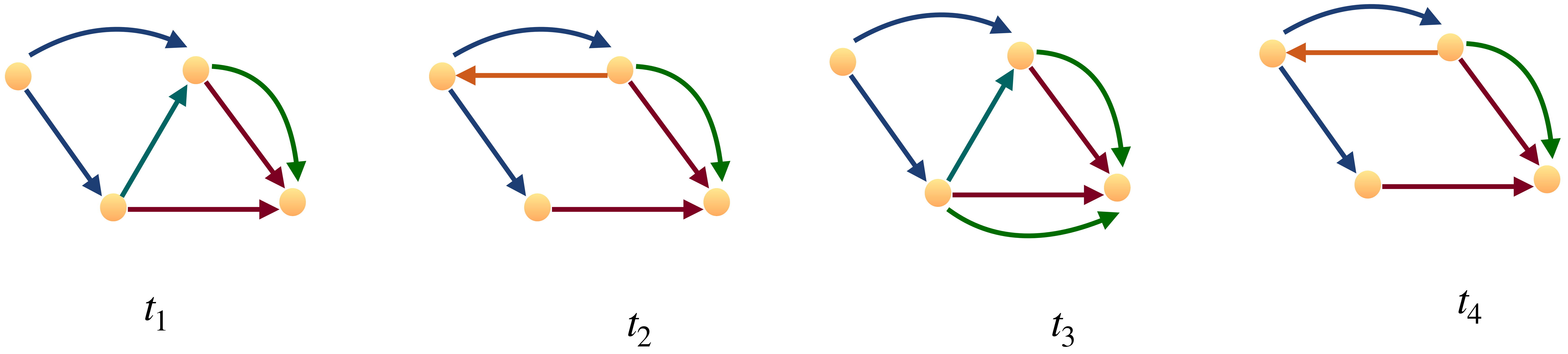
Temporal Knowledge Graphs



Temporal Knowledge Graphs



Temporal Knowledge Graphs

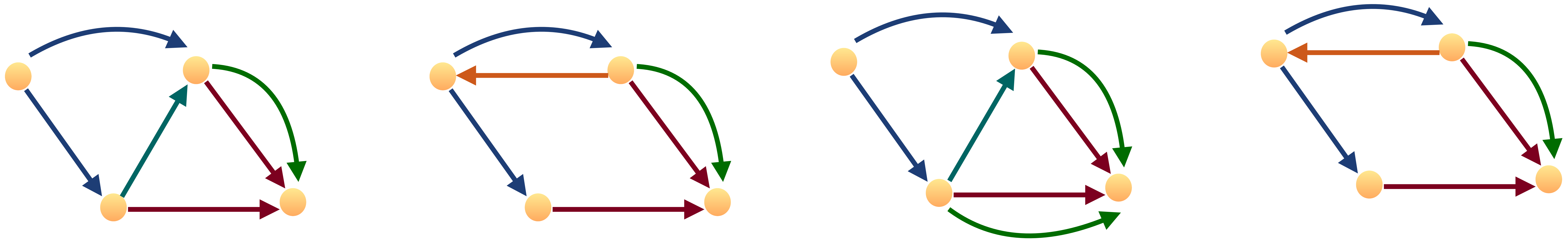


Temporal knowledge: Knowledge changes over time and we can capture this with timestamps T.

Temporal facts: A temporal fact is of the form $r(h, t \mid \tau)$, where $r \in R$, and $h, t \in E$, and $\tau \in T$, indicating the timestamp where the fact holds.

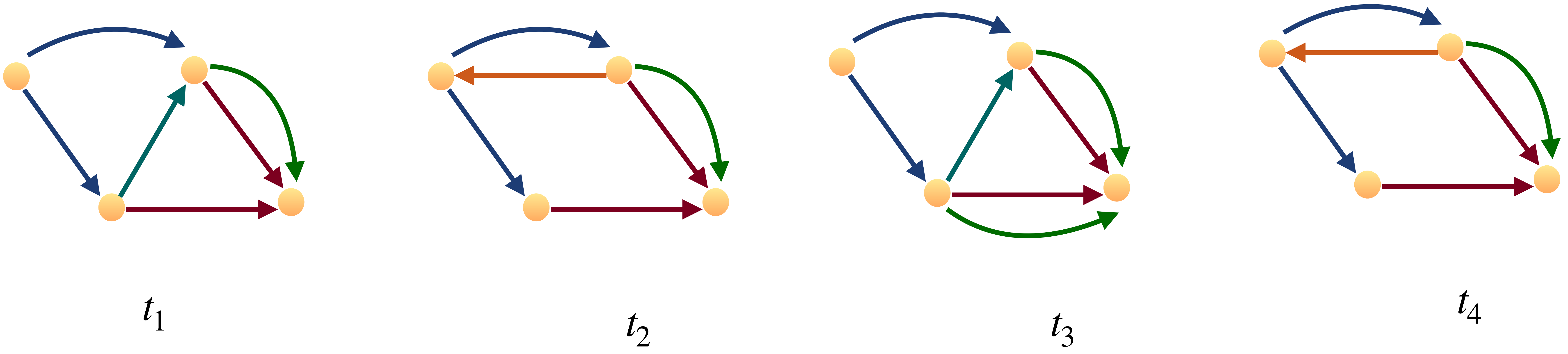
Temporal KGs: A temporal KG is a finite set of temporal facts, or equivalently a sequence of KGs.

Temporal Knowledge Graph Completion



t_1 t_2 t_3 t_4

Temporal Knowledge Graph Completion



Temporal KG completion: Given a temporal KG G, temporal KG completion is to predict (temporal) facts that are missing from G. There are two regimes:

Interpolation: Observations over timestamps $\tau_1 \ldots \tau_n$ and predictions/completion over timestamps $\tau_1 \ldots \tau_n$.

Extrapolation: Observations over timestamps $\tau_1 \ldots \tau_n$ and predictions/completions over unseen timestamps.

Extrapolation is very hard, but already interpolation is hard: facts must be predicted in the right timestamps.

Temporal Knowledge Graph Completion

Dataset	|E|	|R|	|T|	Training	Validation	Test	Timespan	Granularity
ICEWS14	7,128	230	365	72,826	8,963	8,941	1 year	Daily
ICEWS05-15	10,488	251	4017	386,962	46,092	46,275	11 years	Daily
GDELT	500	20	366	2,735,685	341,961	341,961	1 year	Daily

Temporal Knowledge Graph Completion

Dataset	|E|	|R|	|T|	Training	Validation	Test	Timespan	Granularity
ICEWS14	7,128	230	365	72,826	8,963	8,941	1 year	Daily
ICEWS05-15	10,488	251	4017	386,962	46,092	46,275	11 years	Daily
GDELT	500	20	366	2,735,685	341,961	341,961	1 year	Daily

Sampling and evaluation: Sample corrupted facts for each timestamp and evaluate/rank accordingly.

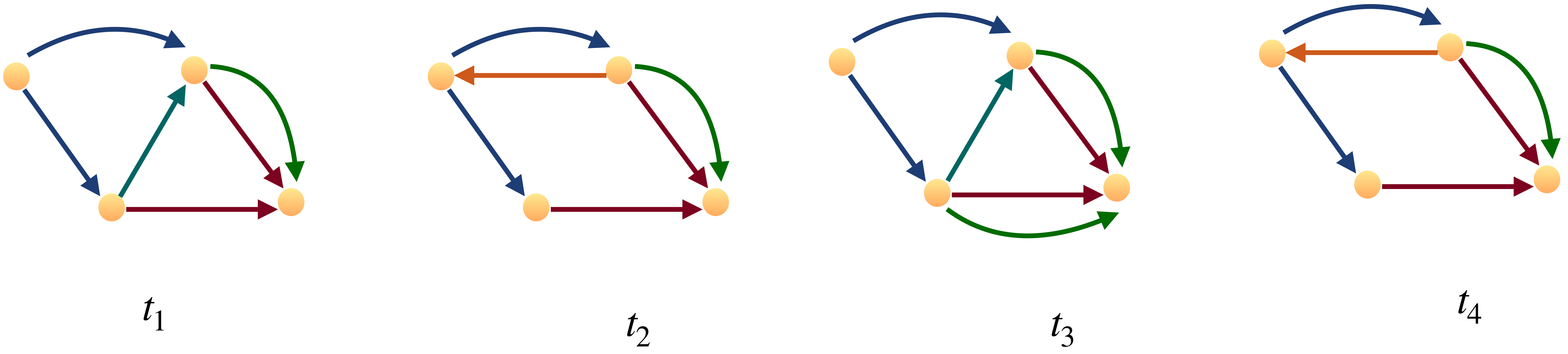
Temporal Knowledge Graph Completion

Dataset	|E|	|R|	|T|	Training	Validation	Test	Timespan	Granularity
ICEWS14	7,128	230	365	72,826	8,963	8,941	1 year	Daily
ICEWS05-15	10,488	251	4017	386,962	46,092	46,275	11 years	Daily
GDELT	500	20	366	2,735,685	341,961	341,961	1 year	Daily

Sampling and evaluation: Sample corrupted facts for each timestamp and evaluate/rank accordingly.

Datasets: ICEWS14 and ICEWS5-15 (Garcia-Duran et al, 2018): subsets of the Integrated Crisis Early Warning System (ICEWS) dataset, which stores temporal socio-political facts starting from the year 1995. GDELT: a subset of Global Database of Events, Language, and Tone temporal KG (Leetaru and Schrodt 2013).

A Brief Look at TTransE



Encoder: TTransE (Leblay and Chekol, 2018) extends TransE by additionally encoding each timestamp $\tau \in \mathbf{T}$ into the space $\boldsymbol{\tau} \in \mathbb{R}^d$.

Decoder: Score a temporal fact $r(h, t \mid \tau)$ based on how similar $\mathbf{h} + \mathbf{r} + \boldsymbol{\tau}$ and $\mathbf{t}$ are: $-\|\mathbf{h} + \mathbf{r} + \boldsymbol{\tau} - \mathbf{t}\|$

Remark: Any translational model can be extended in this simple way and more sophisticated proposals exist.

Outlook and Discussions

Beyond This Lecture

Beyond This Lecture

- **Models**: We focused on representative models, but there are many more...

Beyond This Lecture

- **Models**: We focused on representative models, but there are many more...
  - Beyond Euclidian spaces, e.g., Poincare embeddings.

Beyond This Lecture

- **Models**: We focused on representative models, but there are many more...
  - Beyond Euclidian spaces, e.g., Poincare embeddings.
  - Other geometrical abstractions, e.g., TorusE.

Beyond This Lecture

- **Models**: We focused on representative models, but there are many more...
  - Beyond Euclidian spaces, e.g., Poincare embeddings.
  - Other geometrical abstractions, e.g., TorusE.
- **Practical**: Many regularization/optimization techniques are omitted, see e.g., Rufinelli et al. (2020).

Beyond This Lecture

- **Models**: We focused on representative models, but there are many more...
  - Beyond Euclidian spaces, e.g., Poincare embeddings.
  - Other geometrical abstractions, e.g., TorusE.
- **Practical**: Many regularization/optimization techniques are omitted, see e.g., Rufinelli et al. (2020).
- **Higher-arity knowledge bases**: Real-world data is not necessarily in the form of binary atoms: facts can be of higher arity, e.g., hasDegreeFrom(Hawking,Cambridge,DPhil), and very few models can handle such data.

Beyond This Lecture

- **Models**: We focused on representative models, but there are many more...
  - Beyond Euclidian spaces, e.g., Poincare embeddings.
  - Other geometrical abstractions, e.g., TorusE.
- **Practical**: Many regularization/optimization techniques are omitted, see e.g., Rufinelli et al. (2020).
- **Higher-arity knowledge bases**: Real-world data is not necessarily in the form of binary atoms: facts can be of higher arity, e.g., hasDegreeFrom(Hawking,Cambridge,DPhil), and very few models can handle such data.
- **Rule injection**: KGs usually have an accompanying schema, or an ontology, encoding the general domain knowledge in the form of first-order rules. Ideally, all predictions in the KG completion task should comply with such knowledge. Is it possible to inject such knowledge into the embedding models and to what extent?

Beyond This Lecture

- **Models**: We focused on representative models, but there are many more...
  - Beyond Euclidian spaces, e.g., Poincare embeddings.
  - Other geometrical abstractions, e.g., TorusE.
- **Practical**: Many regularization/optimization techniques are omitted, see e.g., Rufinelli et al. (2020).
- **Higher-arity knowledge bases**: Real-world data is not necessarily in the form of binary atoms: facts can be of higher arity, e.g., hasDegreeFrom(Hawking,Cambridge,DPhil), and very few models can handle such data.
- **Rule injection**: KGs usually have an accompanying schema, or an ontology, encoding the general domain knowledge in the form of first-order rules. Ideally, all predictions in the KG completion task should comply with such knowledge. Is it possible to inject such knowledge into the embedding models and to what extent?
- **Other tasks**: Tasks beyond KG completion, e.g., entity classification, query answering with embedding models.

Summary

- KG completion with shallow embedding models:
  - Translational models, e.g., TransE, RotatE.
  - Bilinear models, e.g., RESCAL, DistMULT, ComplEx.
  - Box embeddings, e.g., BoxE.
- Many other embedding models build on similar, or analogous ideas.
- Temporal KG completion
- We evaluated the respective models in terms of:
  - Model expressiveness
  - Model inductive capacity and inference patterns

References

- A. Bordes, J. Weston, R. Collobert, and Y. Bengio, Learning structured embeddings of knowledge bases. *AAAI*, 2011.
- A. Bordes, X. Glorot, J. Weston, and Y. Bengio, Joint learning of words and meaning representations for open-text semantic parsing. *AISTATS*, 2012.
- A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *NIPS*, 2013.
- A. Bordes, X. Glorot, J. Weston, and Y. Bengio. A semantic matching energy function for learning with multi-relational data. *Mach. Learn.*, 2014.
- M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. *ICML*, 2011.
- T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction". *ICML*, 2016.
- I. Balazevic, C. Allen, and T. Hospedales. TuckER: Tensor factorization for knowledge graph completion. *EMNLP-IJCNLP*, 2019.
- Z. Wang, J. Zhang, J. Feng, and Z. Chen, Knowledge graph embedding by translating on hyperplanes. *AAAI*, 2014.
- S. He, K. Liu, G. Ji, and J. Zhao. Learning to represent knowledge graphs with Gaussian embedding. *CIKM*, 2015.
- R. Socher, D. Chen, C. D. Manning, and A. Y. Ng. Reasoning with neural tensor networks for knowledge base completion. *NIPS*, 2013.

References

- R. Abboud, İ.İ. Ceylan, T.Lukasiewicz, T. Salvatori. BoxE: A Box Embedding Model for Knowledge Base Completion. *NeurIPS*, 2020.
- L. Vilnis, X. Li, X., S. Murty, and A. McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. *ACL, 2018*.
- H. Ren, W. Hu, J. Leskovec. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings, *ICLR*, 2020.
- M. Schlichtkrull, T. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modelling Relational Data with Graph Convolutional Networks. *ESWC*, 2018.
- M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. *NIPS*, 2017.
- D. Ruffinelli, S. Broscheit, R. Gemulla. You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings, *ICLR*, 2020.
- B. Yang, W.-T. Yih, X. He, J. Gao, and L. Deng, Embedding entities and relations for learning and inference in knowledge bases. *ICLR*, 2015.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.
- Z. Sun, Z. Deng, J. Nie, and J. Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. *ICLR*, 2019.

References

- L. Cai and W. Y. Wang. KBGan: Adversarial learning for knowledge graph embeddings. *NAACL-HLT*, 2018.
- T. Ebisu and R. Ichise. TorusE: Knowledge graph embedding on a lie group. *AAAI*, 2018.
- V. Gutiérrez-Basulto and S. Schockaert. From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules. *KR*, 2018
- W.L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. *NeurIPS*, 2017.
- T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel. Convolutional 2D knowledge graph embeddings. *AAAI*, 2018.
- Garcia-Duran, A.; Dumancic, S.; and Niepert, M. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In EMNLP, 2018.
- Leetaru, K.; and Schrodt, P. A. 2013. GDELT: Global data on events, location, and tone. ISA Annual Convention.
- Leblay, J. and Chekol, M. W. (2018). Deriving validity time in knowledge graph. WWW, 2018.