Embedding Techniques

MATRICES, TENSORS, AND NEURAL NETWORKS

Graphical Models: Downsides

Embeddings

Limitation to Logical Relations

- Representation restricted by manual design
  - Clustering? Assymetric implications?
  - Information flows through these relations
- Difficult to generalize to unseen entities/relations

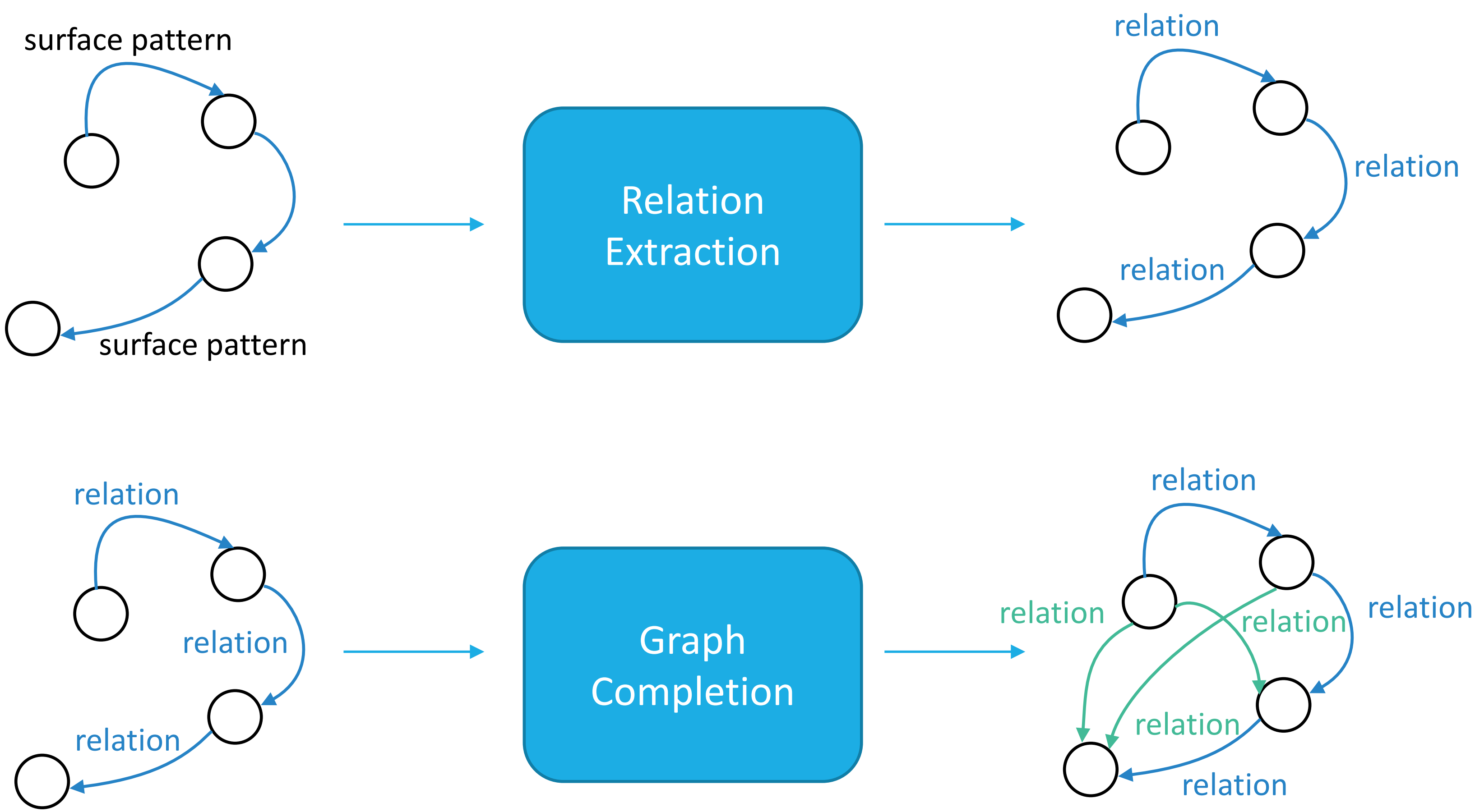
- Everything as dense vectors
- Captures many relations
- Learned from data

Computational Complexity of Algorithms

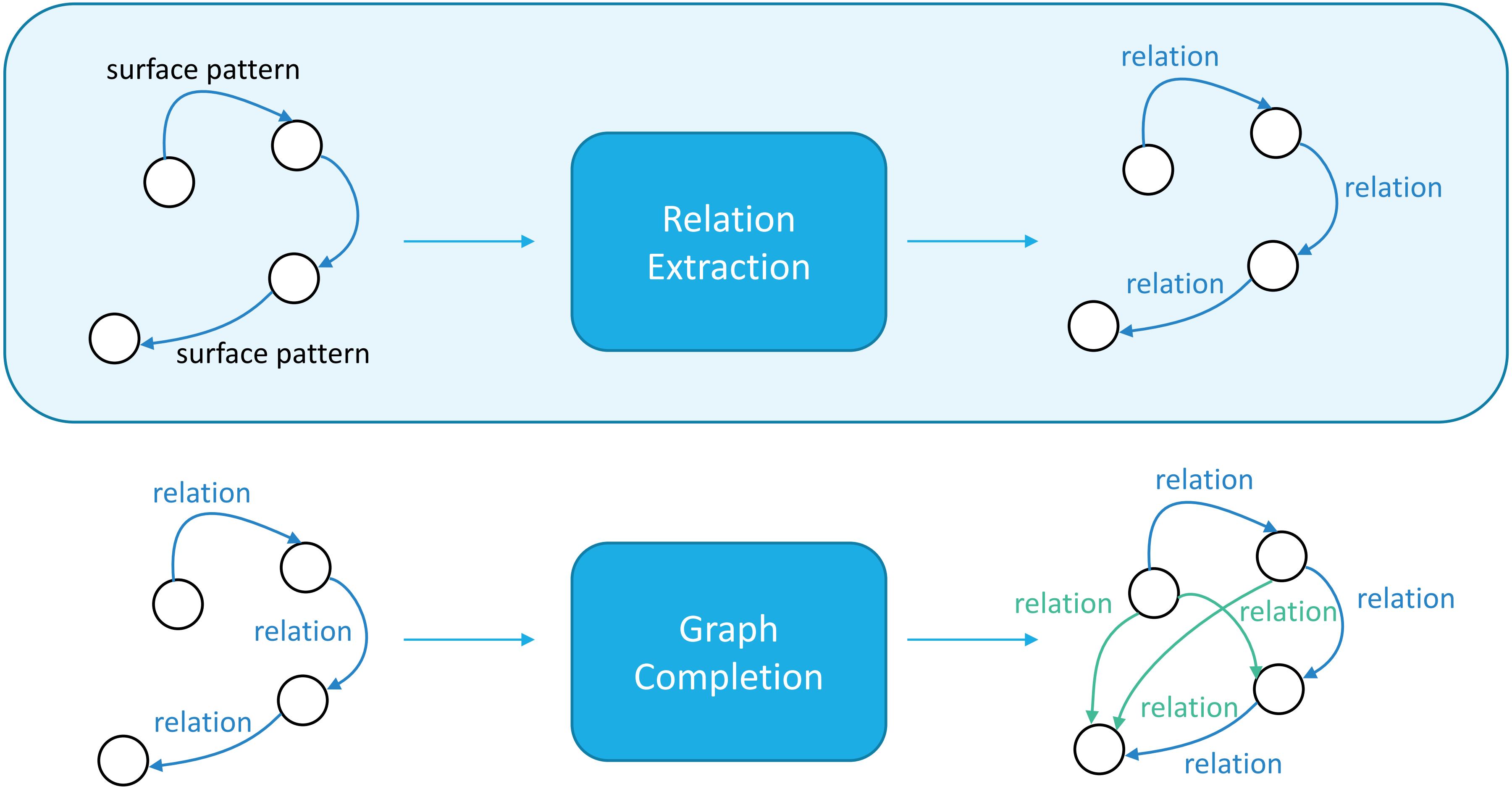
- Learning is NP-Hard, difficult to approximate
- Query-time inference is also NP-Hard
- Not easy to parallelize, or use GPUs
- Scalability is badly affected by representation

- Learning using stochastic gradient, back-propagation
- Querying is often cheap
- GPU-parallelism friendly

Two Related Tasks



Two Related Tasks



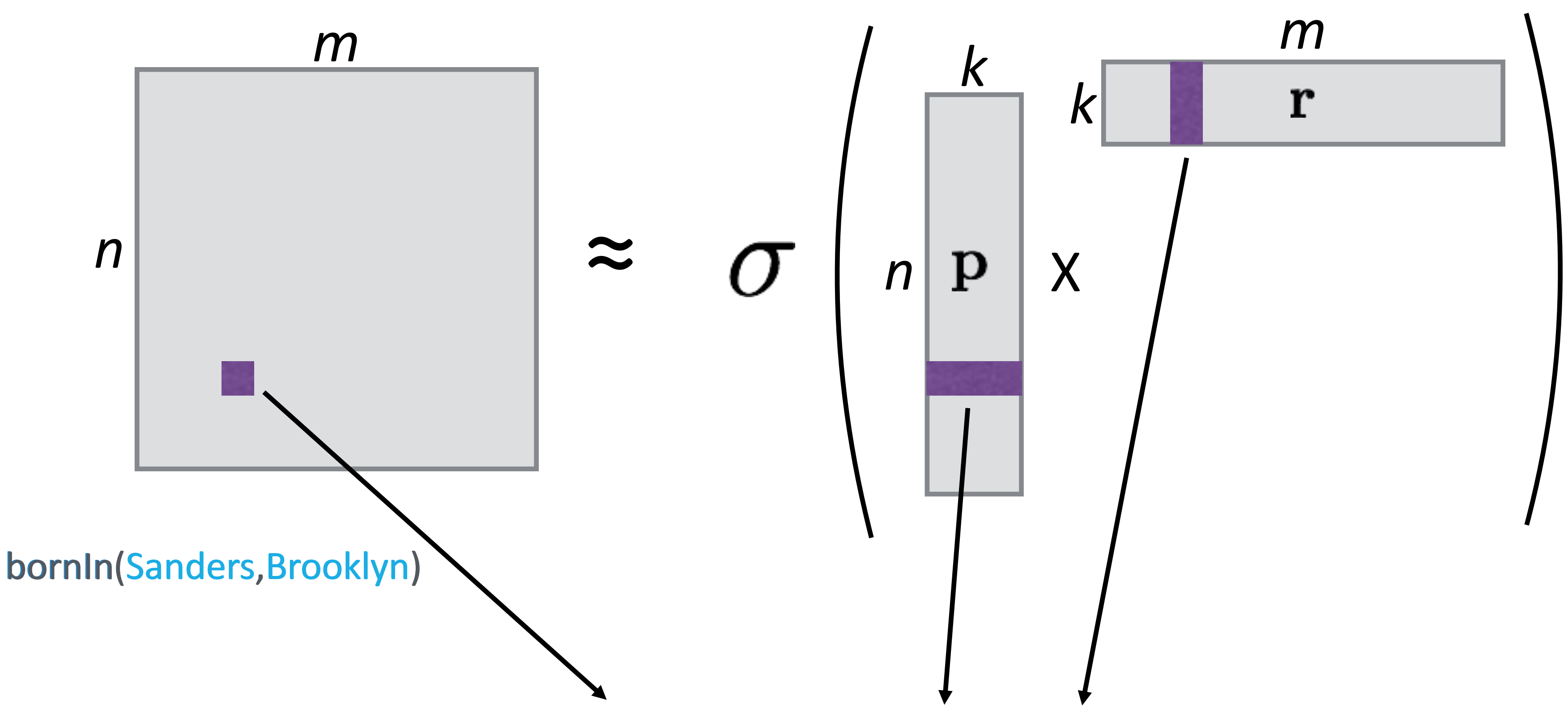
Relation Extraction as a Matrix

Sanders was born in Brooklyn, to Dorothy and Eli Sanders.

Entity Pairs	was born in <-nsubjpass-born<-nmod:in-	was born to	and	birthplace(X,Y)	spouse(X,Y)
Bernie Sanders, Brooklyn	1			?	
Bernie Sanders, Dorothy Sanders		1			
Bernie Sanders, Eli Sanders		1			
Dorothy Sanders, Eli Sanders			1		?
Barack Obama, Hawaii	1			1	
Barack Obama, Michelle Obama			1		1

[Riedel et al NAACL 2013]

Matrix Factorization



$$P(R(i,j)) = \sigma(\mathbf{p}_{i,j} \cdot \mathbf{r}_R)$$

Training

Loss Function:

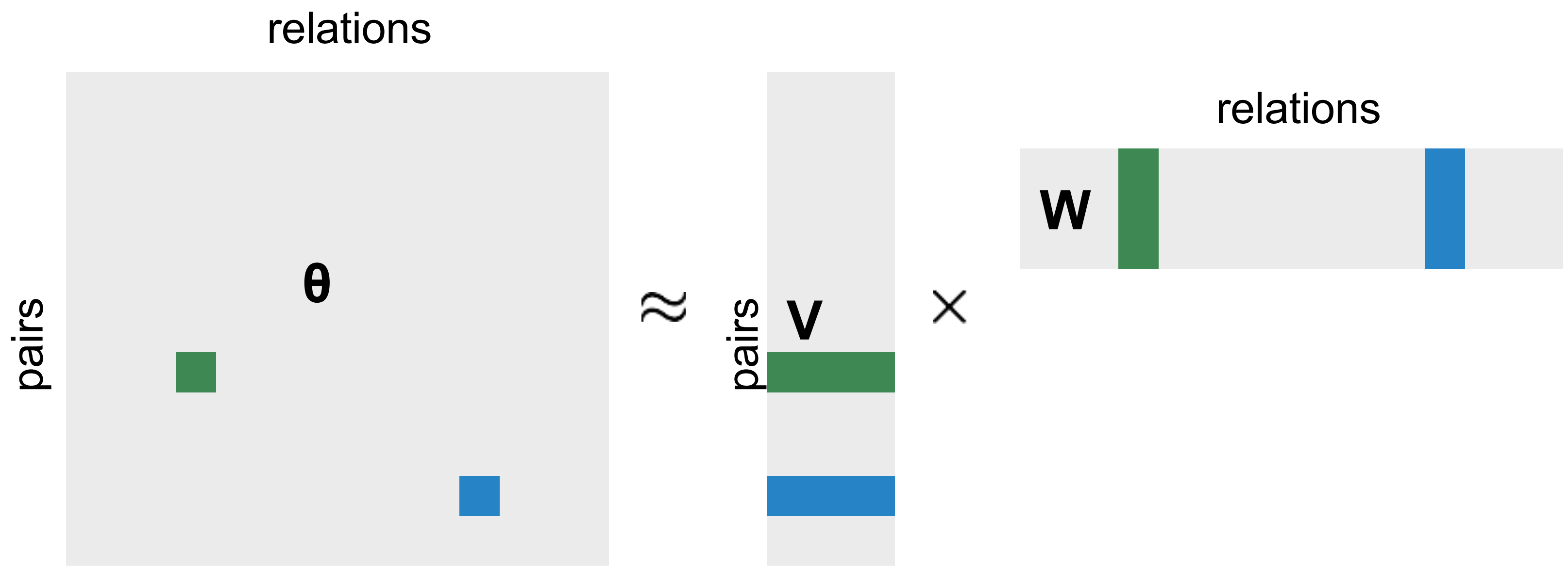
$$\max_{\mathbf{v},\mathbf{w}} \log \prod_{x,y,r} \exp\langle \mathbf{v}^{x,y}, \mathbf{w}_r \rangle - \lambda(||\mathbf{v}||_2^2 + ||\mathbf{w}||_2^2)$$

Desiderata from the training algorithm:

- Do not instantiate the whole matrix!
- Do not hold all the observed cells in memory
- Each iteration linear in the no. of observations

Solution: Stochastic Gradient Descent!

Training: Stochastic Updates



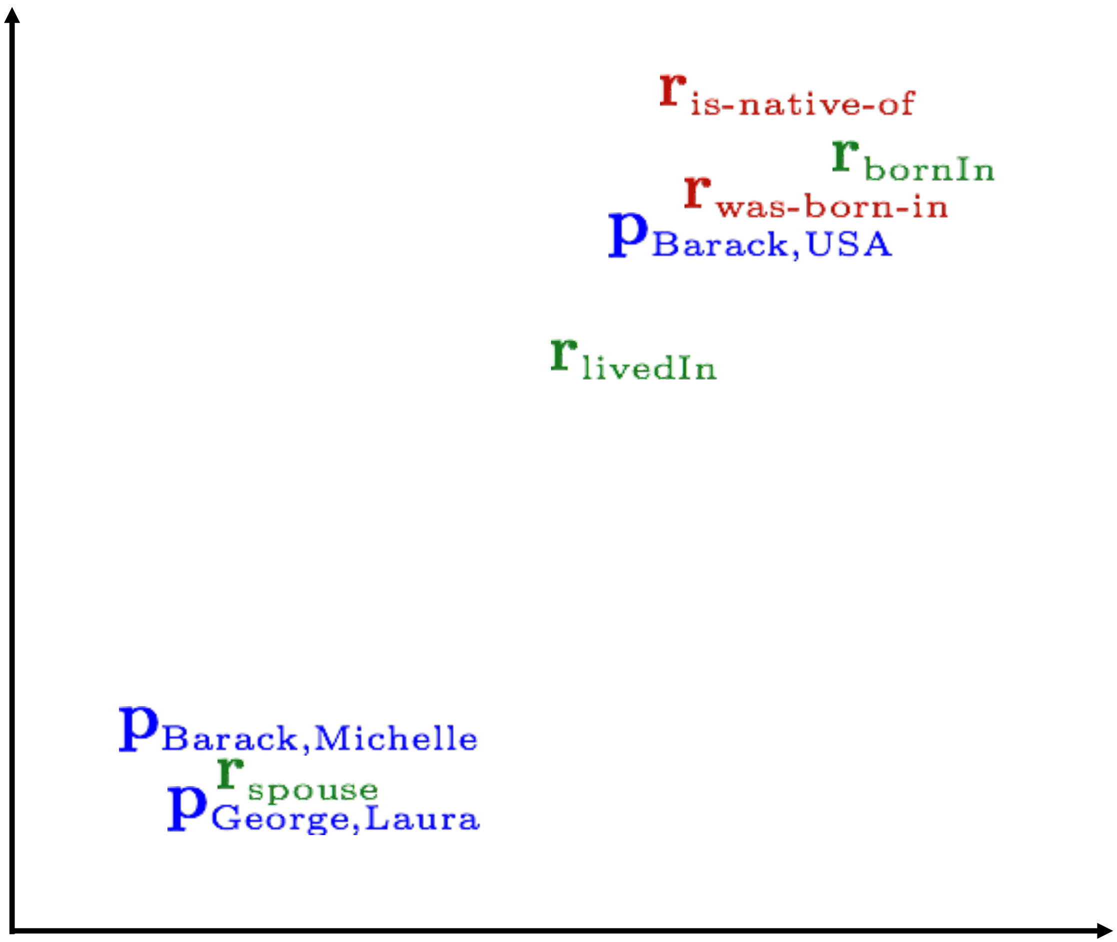
Pick an observed cell, $\theta^r_{x,y}$:

- Update $\mathbf{v}^{x,y}$ & $\mathbf{w}^r$ such that $\theta^r_{x,y}$ is higher

Pick any random cell, assume it is negative:

- Update $\mathbf{v}^{x,y}$ & $\mathbf{w}^r$ such that $\theta^r_{x,y}$ is lower

Relation Embeddings



Embeddings ~ Logical Relations

Relation Embeddings, w

- Similar embedding for 2 relations denote they are paraphrases
  - is married to, spouseOf(X,Y), /person/spouse
- One embedding can be contained by another
  - w(topEmployeeOf) ⊂ w(employeeOf)
  - topEmployeeOf(X,Y) → employeeOf(X,Y)
- Can capture logical patterns, without needing to specify them!

Entity Pair Embeddings, v

- Similar entity pairs denote similar relations between them
- Entity pairs may describe multiple “relations”
  - independent foundedBy and employeeOf relations

Similar Embeddings

similar underlying embedding

similar embedding

	X **own percentage of** Y	X **buy stake in** Y
Time, Inc Amer. Tel. and Comm.	**1**	**1**
Volvo Scania A.B.		**1**
Campeau Federated Dept Stores		
Apple HP		

Successfully predicts “Volvo owns percentage of Scania A.B.” from “Volvo bought a stake in Scania A.B.”

Implications

X historian at Y → X professor at Y

	X **professor at** Y	X **historian at** Y
Kevin Boyle Ohio State		1
R. Freeman Harvard	1	

(Freeman,Harvard)
→ (Boyle,OhioState)

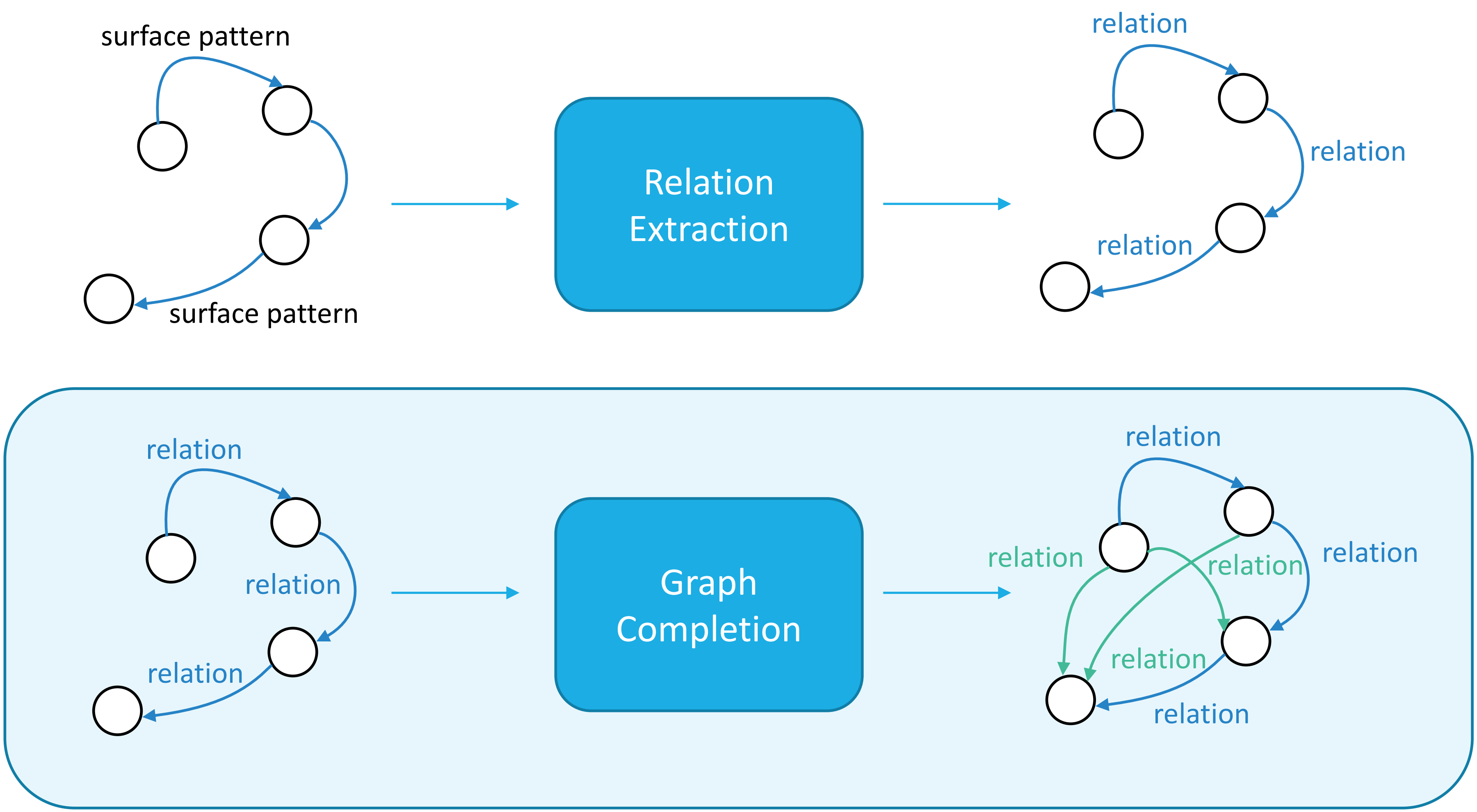
Learns asymmetric entailment:

PER historian at UNIV → PER professor at UNIV

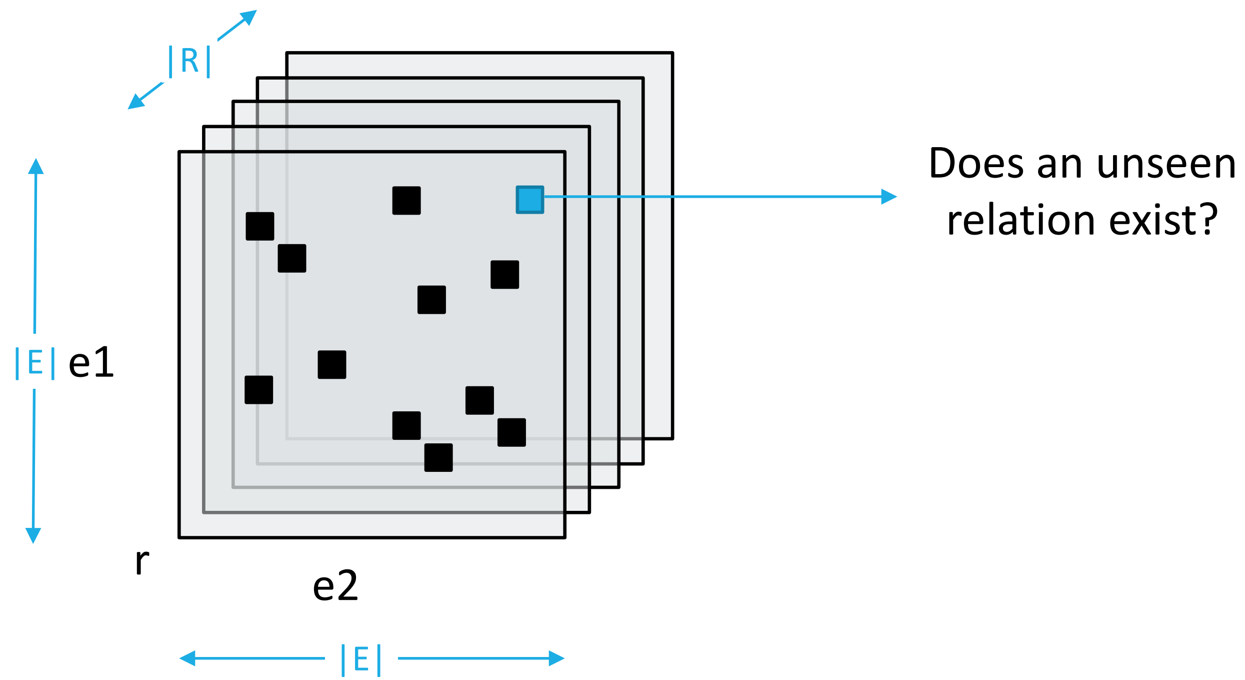
But,

PER professor at UNIV $\not\rightarrow$ PER historian at UNIV

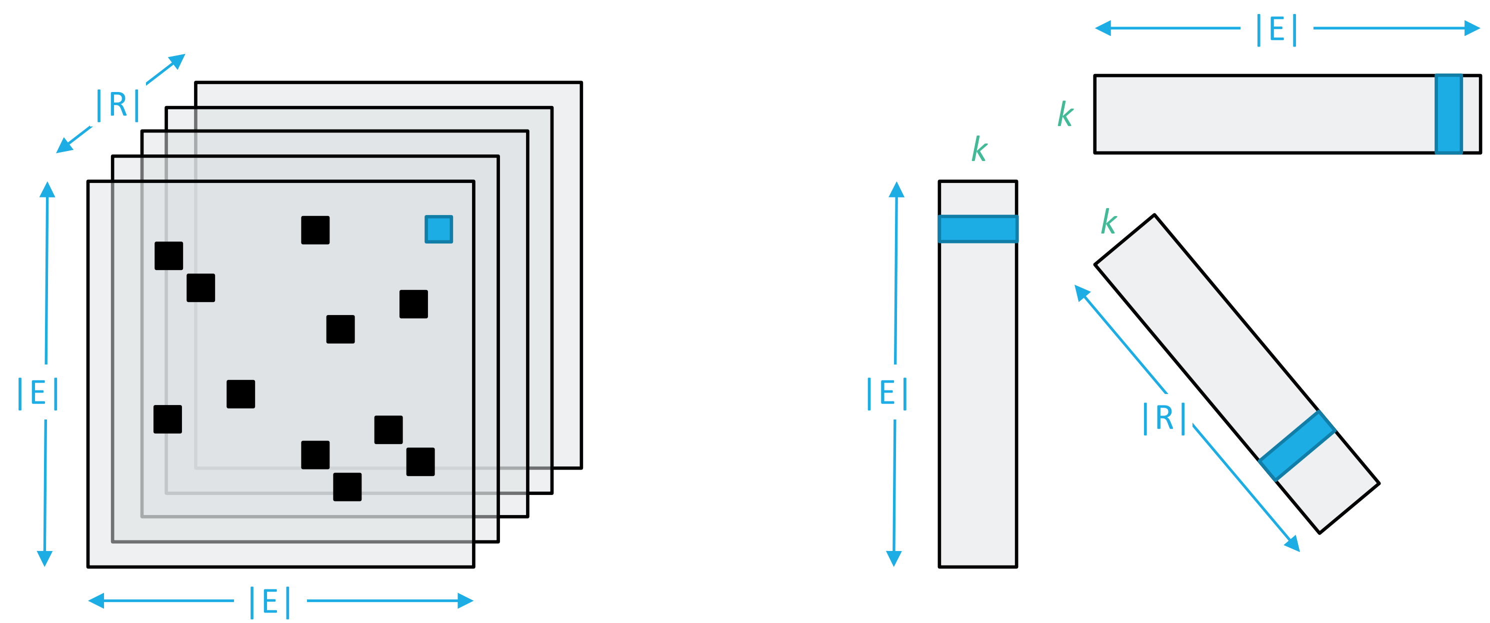
Two Related Tasks



Tensor Formulation of KG



Factorize that Tensor



Many Different Factorizations

CANDECOMP/PARAFAC-Decomposition

$$S\left(r(a,b)\right) = \sum_k R_{r,k} \cdot e_{a,k} \cdot e_{b,k}$$

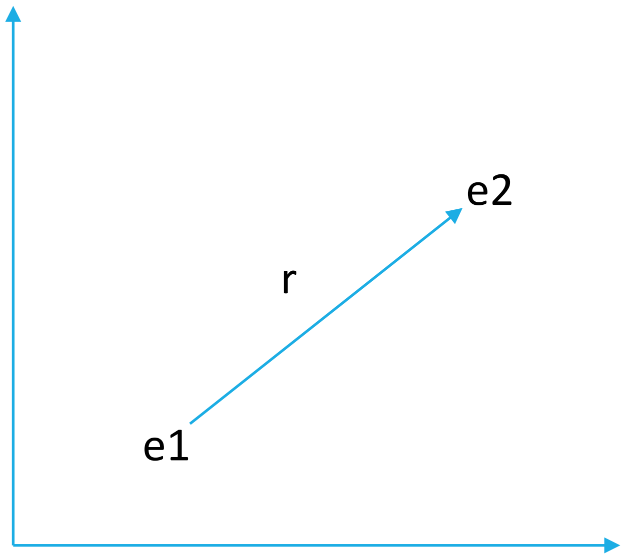
Tucker2 and RESCAL Decompositions

$$S\left(r(a,b)\right) = \left(\mathbf{R}_r \times \mathbf{e}_a\right) \times \mathbf{e}_b$$

Model E

$$S\left(r(a,b)\right) = \mathbf{R}_{r,1} \cdot \mathbf{e}_a + \mathbf{R}_{r,2} \cdot \mathbf{e}_b$$

Translation Embeddings



TransE

$$S\left(r(a,b)\right) = -\|\mathbf{e}_a + \mathbf{R}_r - \mathbf{e}_b\|_2^2$$

Parameter Estimation: SGD

Training Objective

$$\theta = \operatorname*{argmax}_{\theta} \sum_{r_{ab} \in \mathcal{P}} \sum_{r'_{a'b'} \in \mathcal{N}} \mathcal{L}()$$

Distance

$$\mathcal{L}\left(x, y\right) = -\|x - y\|_2^2$$

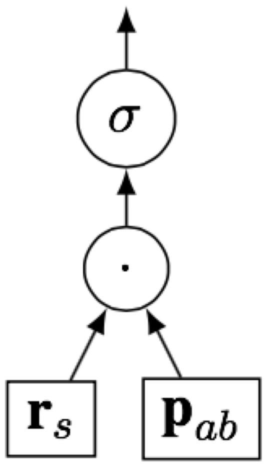
Likelihood

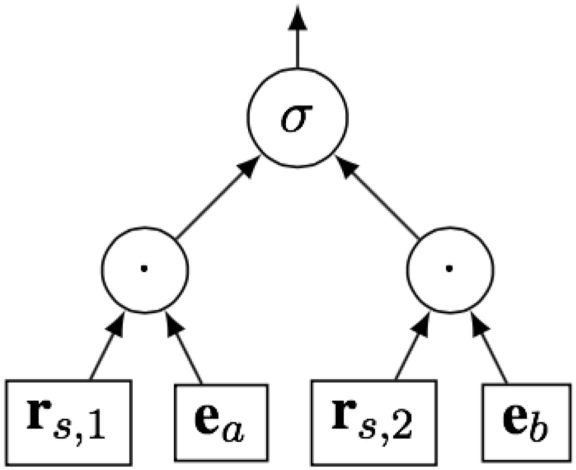
$$\mathcal{L}\left(x, y\right) = p(x)^{p(y)}\left(1 - p(x)\right)^{(1 - p(y))}$$

Stochastic Gradient Descent

Negative Sampling …

Matrix vs Tensor Factorization

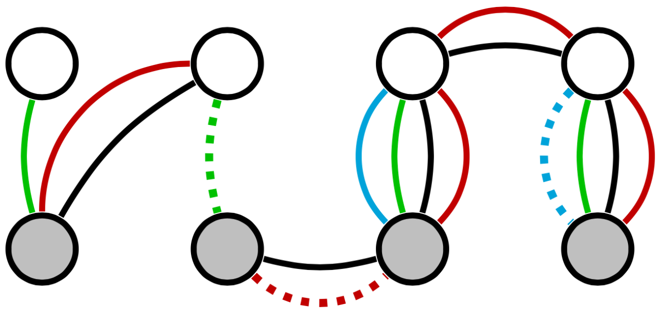


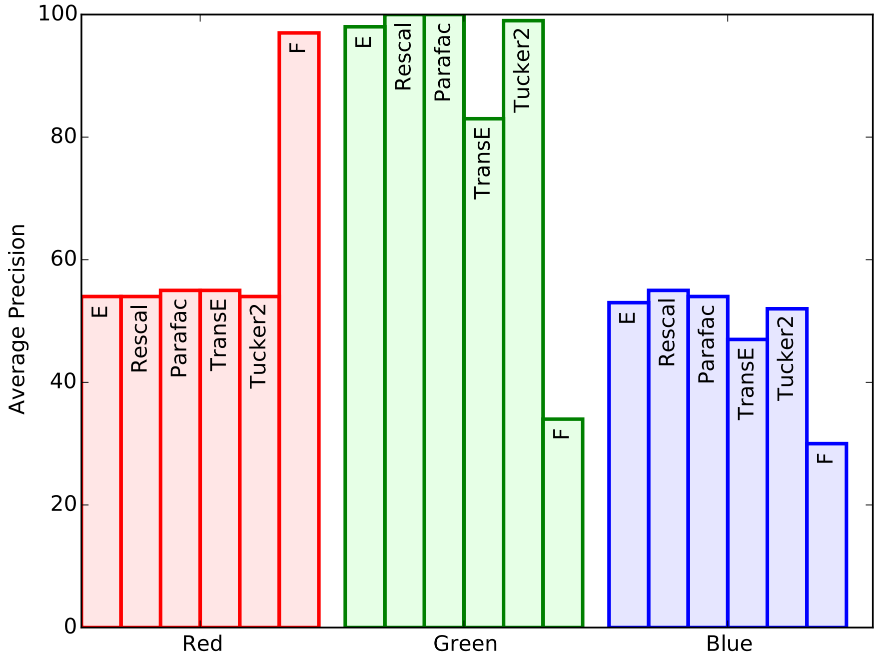


- No encoding of type information
- Can only predict for entity pairs that appear in text together
- Sufficient evidence has to be seen for each entity pair

- Assume low-rank for pairs
- But many relations are not!
- Spouse: you can have only ~1
- Cannot learn pair specific information

What they can, and can't, do..





- **Red**: deterministically implied by **Black**
  - needs *pair-specific* embedding
  - Only **F** is able to generalize
- **Green**: needs to estimate entity types
  - needs *entity-specific* embedding
  - Tensor factorization generalizes, **F** doesn't
- **Blue**: implied by **Red** and **Green**
  - Nothing works much better than random

Compositional Neural Models

So far, we're learning vectors for each entity/surface pattern/relation..

But learning vectors independently ignores "composition"

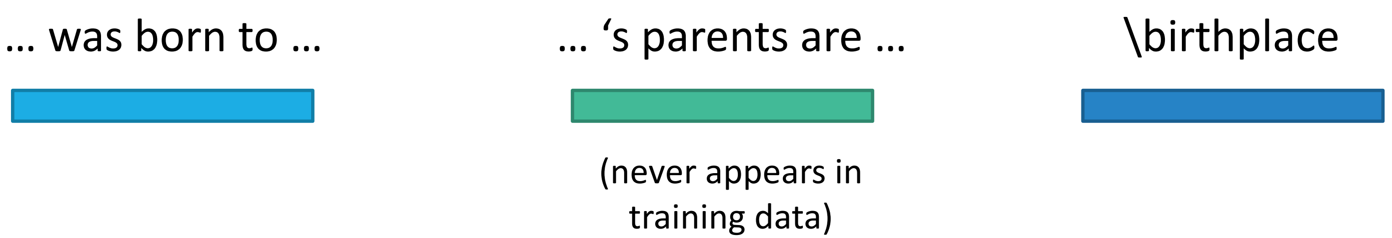
Composition in Surface Patterns

- Every surface pattern is not unique
- Synonymy: A is B's spouse.
  A is married to B.
- Inverse: X is Y's parent.
  Y is one of X's children.
- Can the representation learn this?

Composition in Relation Paths

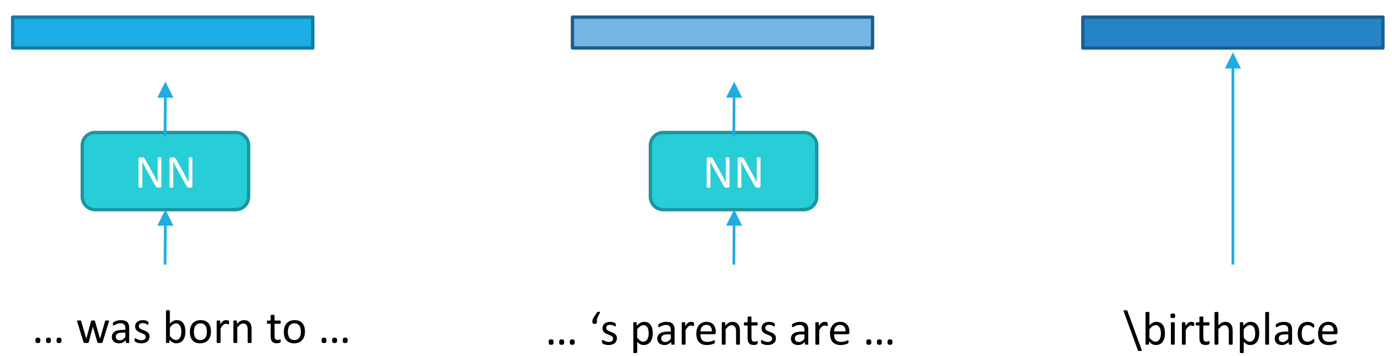
- Every relation path is not unique
- Explicit: A parent B, B parent C
  A grandparent C
- Implicit: X bornInCity Y, Y cityInState Z
  X "bornInState" Z
- Can the representation capture this?

Composing Dependency Paths



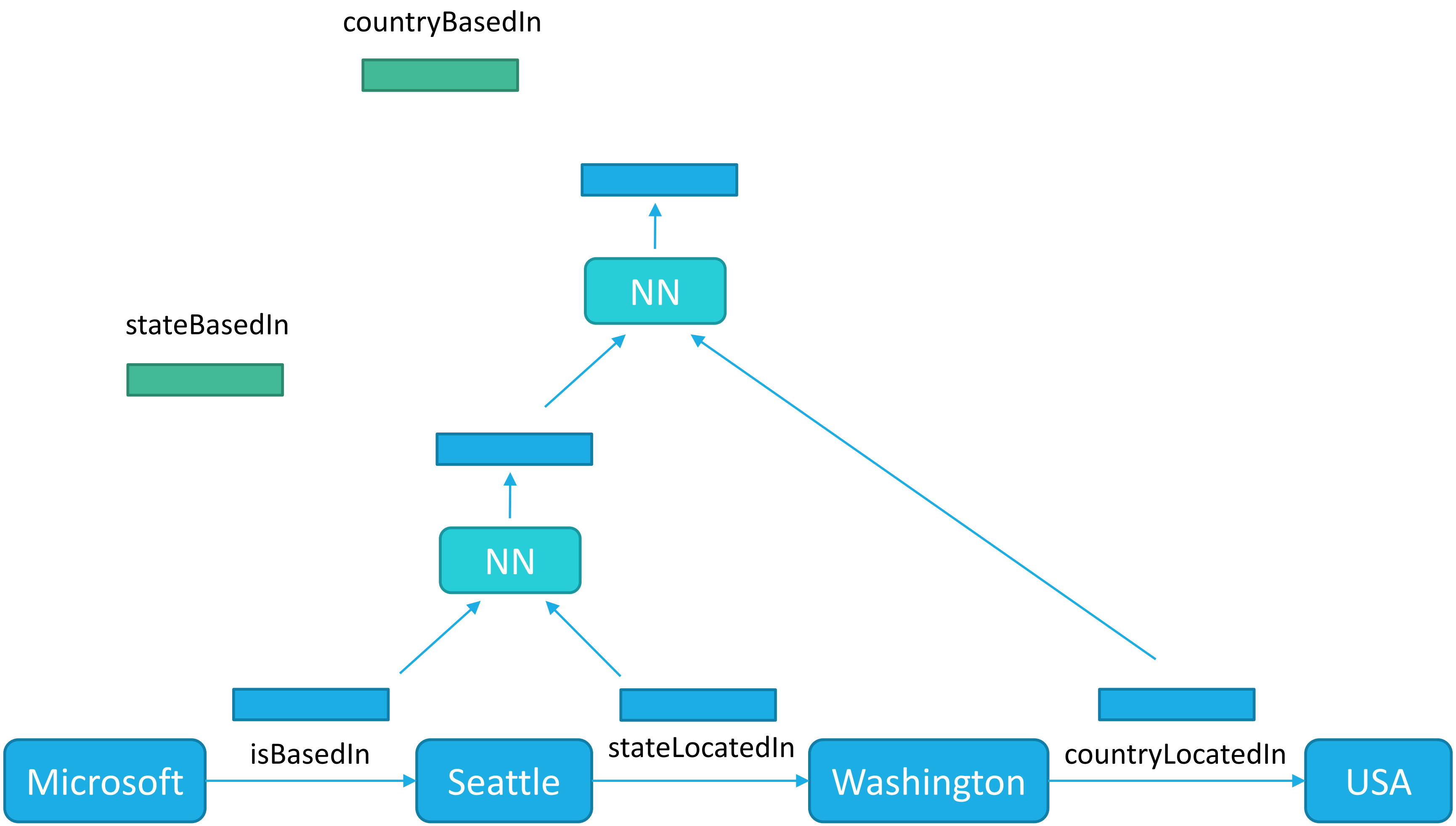
But we don’t need linked data to know they mean similar things…

Use neural networks to produce the embeddings from text!



Verga et al (2016), https://arxiv.org/pdf/1511.06396v2.pdf

Composing Relational Paths



Review: Embedding Techniques

Two Related Tasks:

- Relation Extraction from Text
- Graph (or Link) Completion

Relation Extraction:

- Matrix Factorization Approaches

Graph Completion:

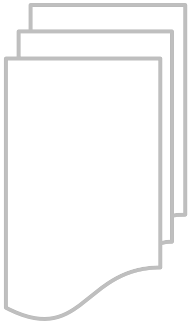
- Tensor Factorization Approaches

Compositional Neural Models

- Compose over dependency paths
- Compose over relation paths

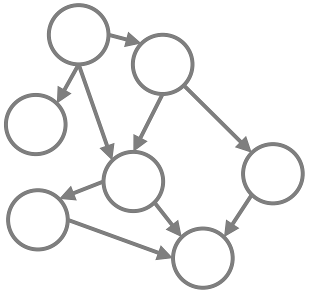
Tutorial Overview

Part 1: Knowledge Graphs



Part 2: Knowledge Extraction

Part 3: Graph Construction



Part 4: Critical Analysis