Knowledge Graph Embeddings

Basel Shbita

INF 558/CSCI 563: Building Knowledge Graphs

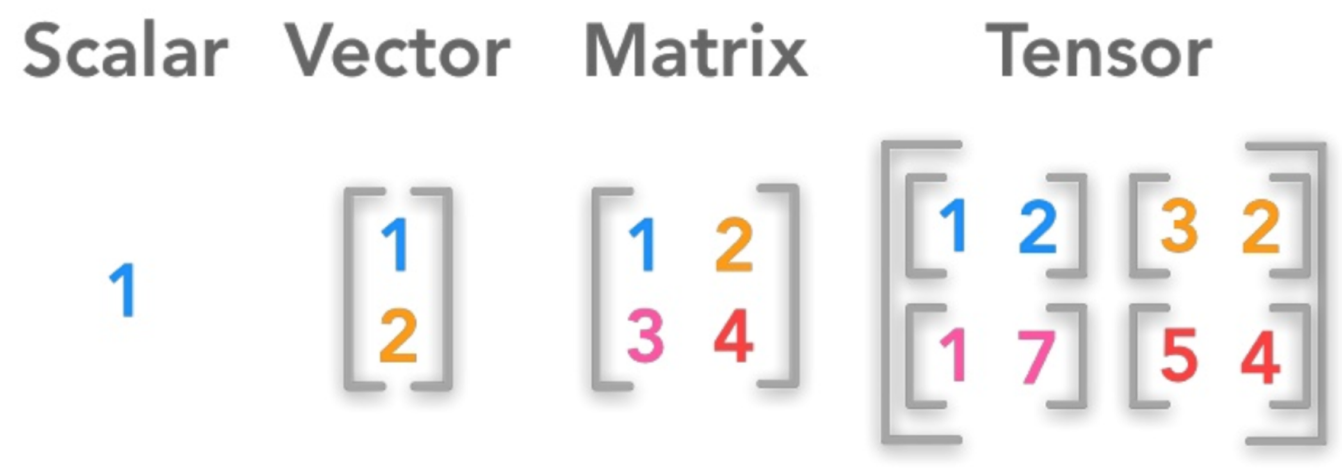
Spring 2020, University of Southern California

* **Some of the slides were provided by:** Jay Pujara, Mayank Kejriwal, Luna Dong, Christos Faloutsos, Andrey Kan, Jun Ma, Subho Mukherjee, Sebastian Riedel, Antoine Bordes

Agenda

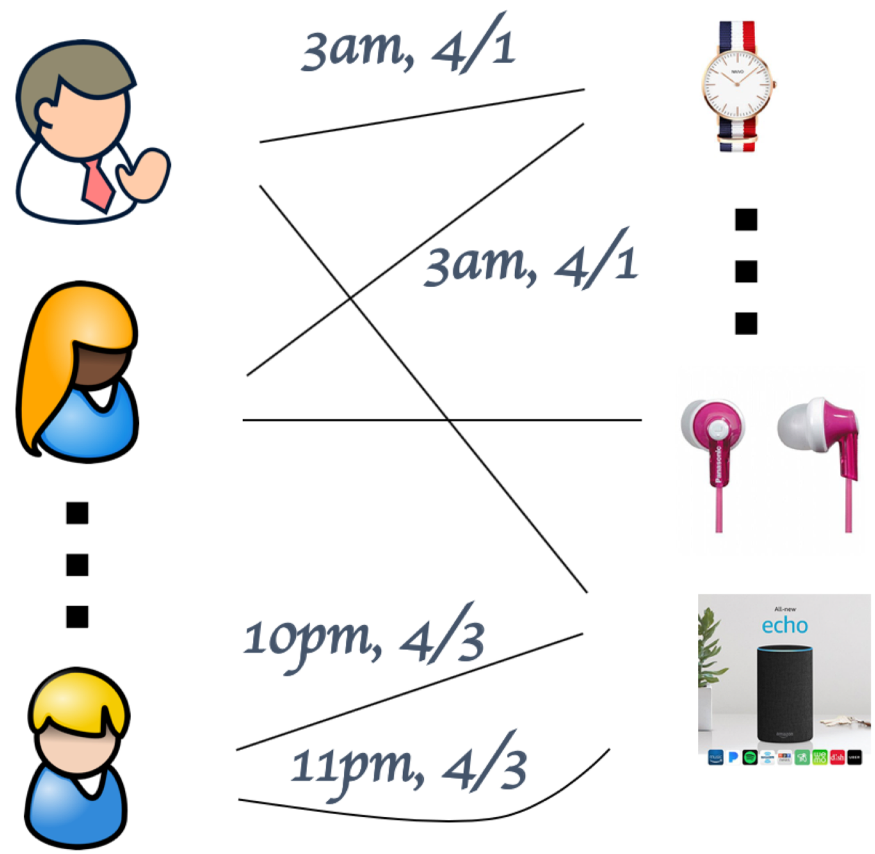
- Motivation
  - Tensors
  - Graphs
  - Embeddings
  - Problem Definition
- Graph Embedding
- Tensor Embedding
- Knowledge Graph Embedding

Tensors



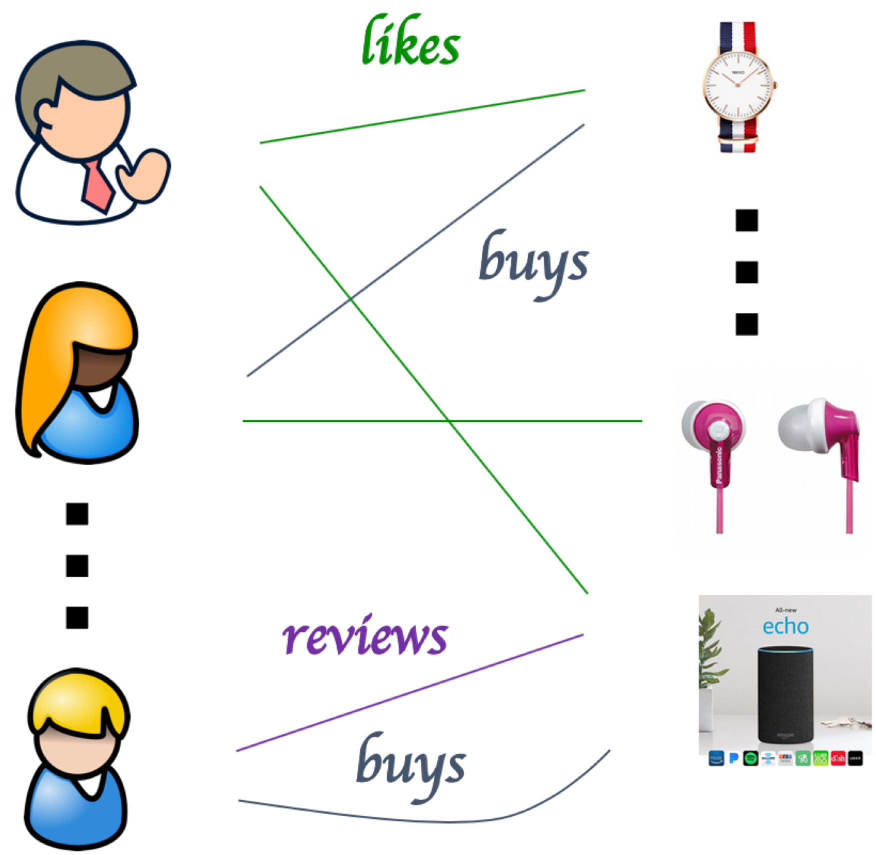
Tensors

- e.g., Time-evolving graphs
- What is ‘normal’? ‘suspicious’?
  - Groups?



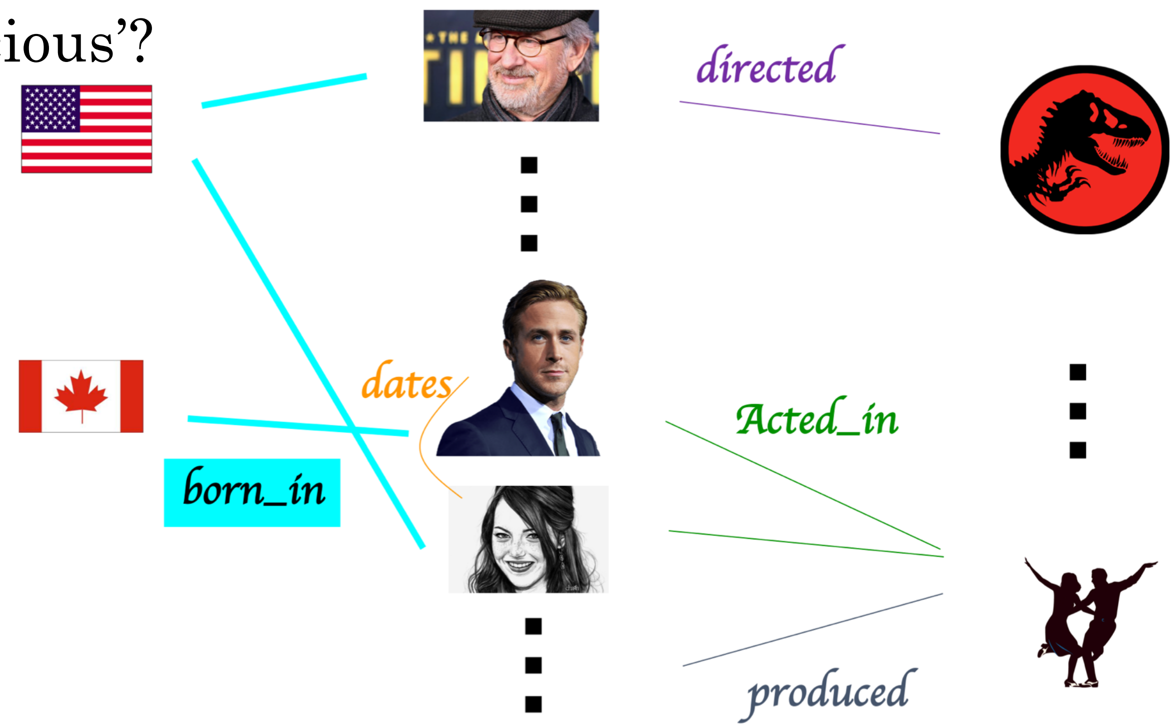
Tensors

- e.g., MultiView Graph
- What is 'normal'? 'suspicious'?
  - Groups?



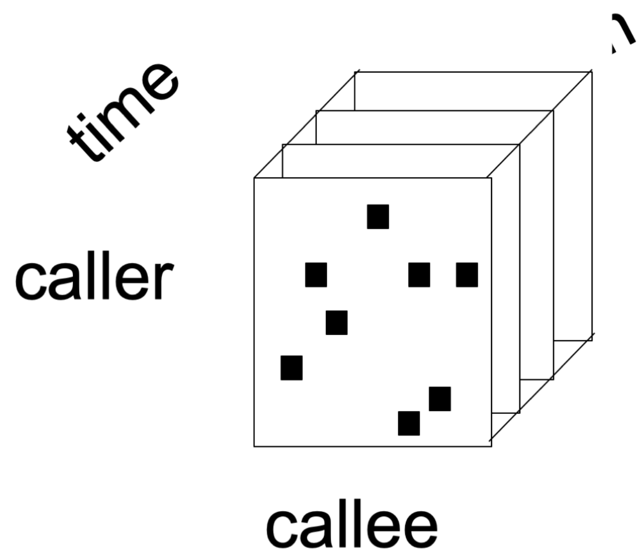
Tensors

- e.g., Knowledge Graphs
- What is 'normal'? 'suspicious'?
  - Groups?



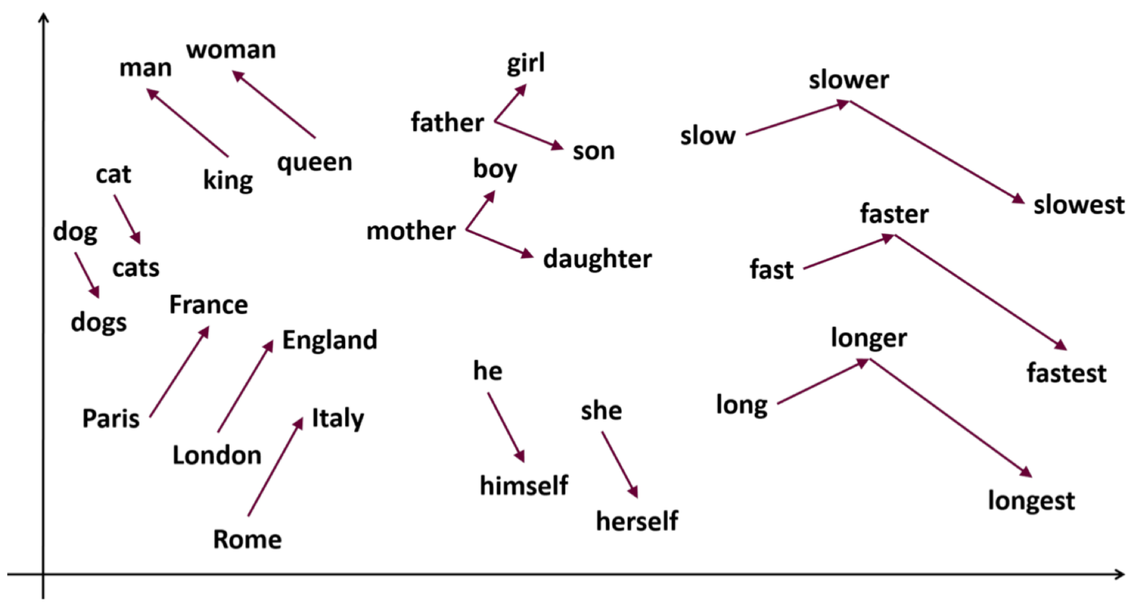
Graphs over time -> tensors!

- Problem #1:
  - Given who calls whom, and when
  - Find patterns / anomalies



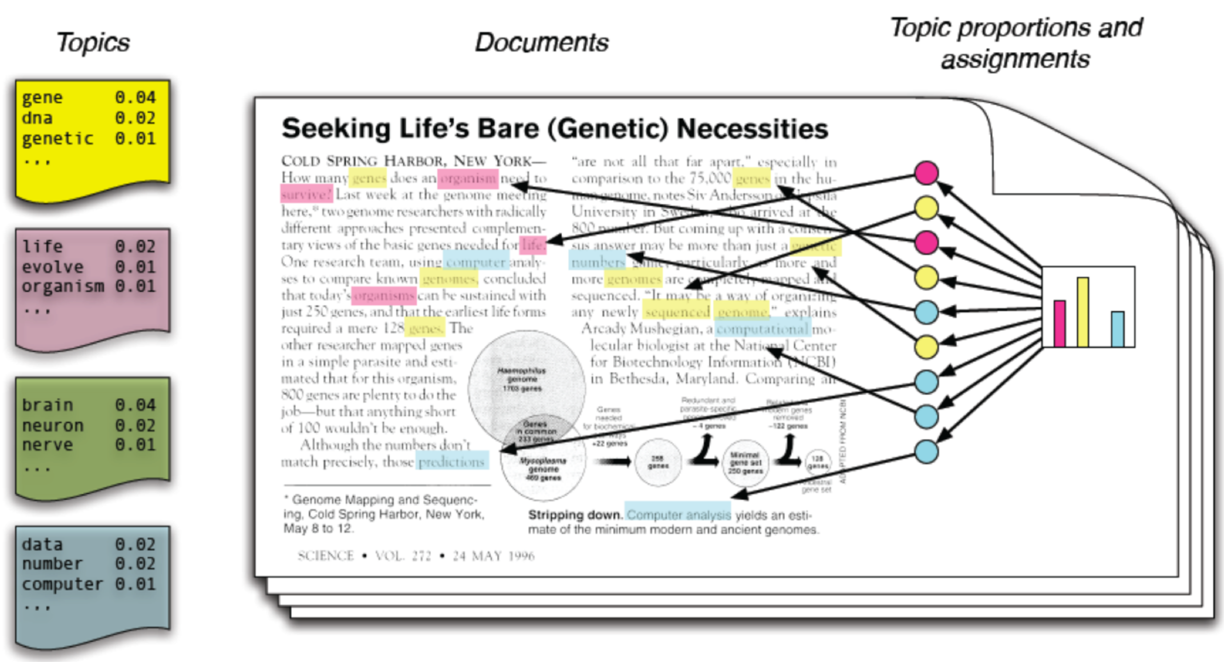
Embedding

- Mapping of discrete variable to a vector of continuous numbers
- Low-dimensional
- Very popular for documents, graphs, words…



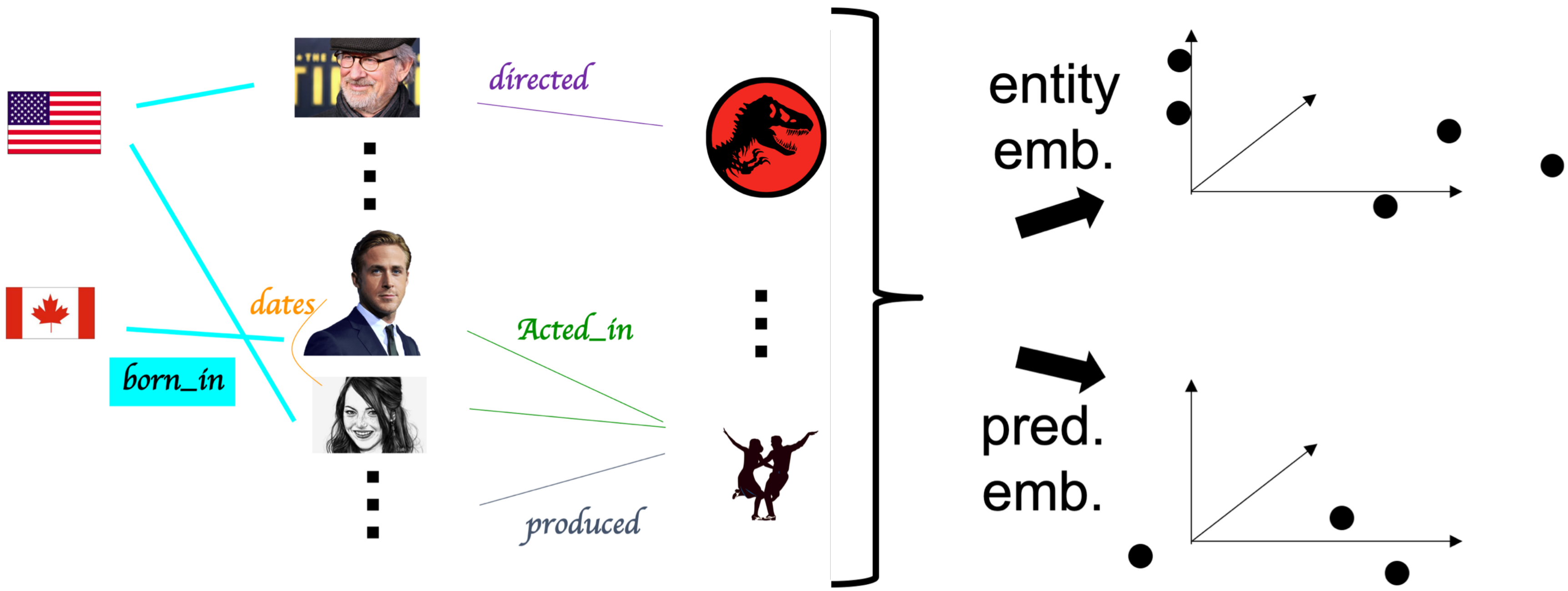
Embedding

- Embeddings are not a ‘new’ invention… topic models are an early example still widely used



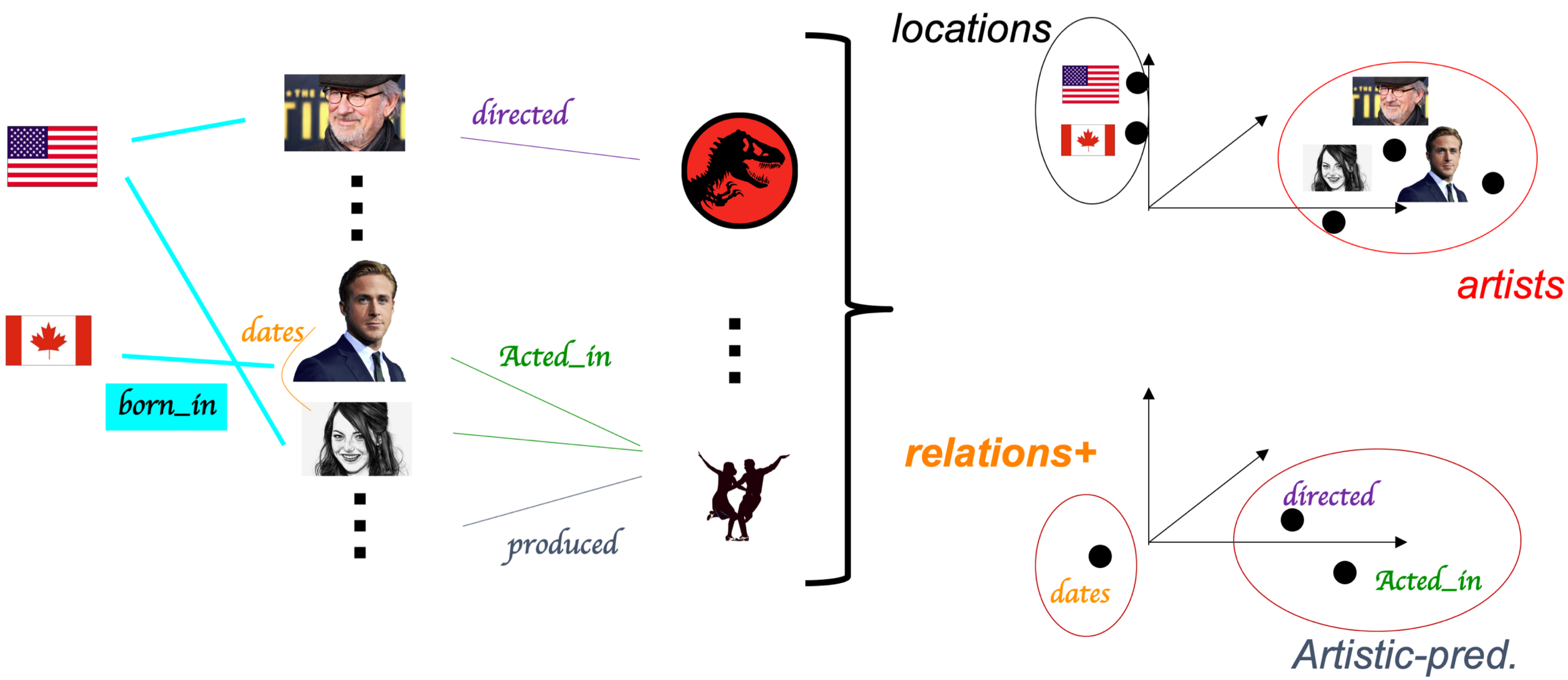
Problem Definition

- Given entities & predicates, find mappings



Problem Definition

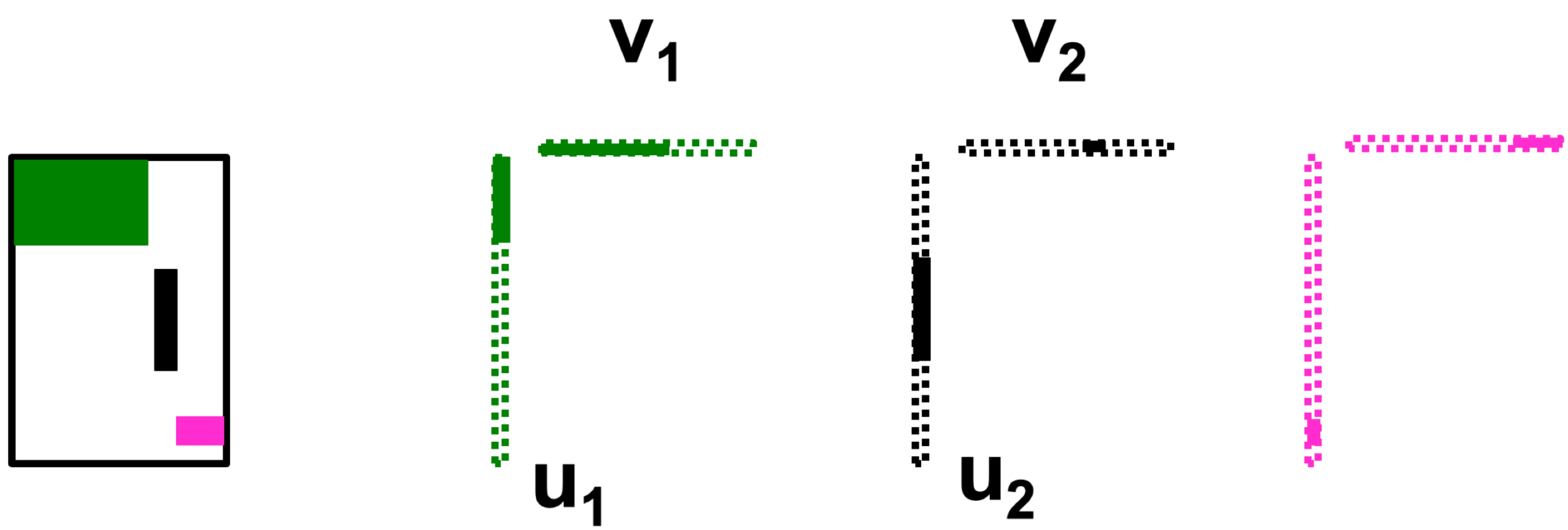
- Given entities & predicates, find mappings



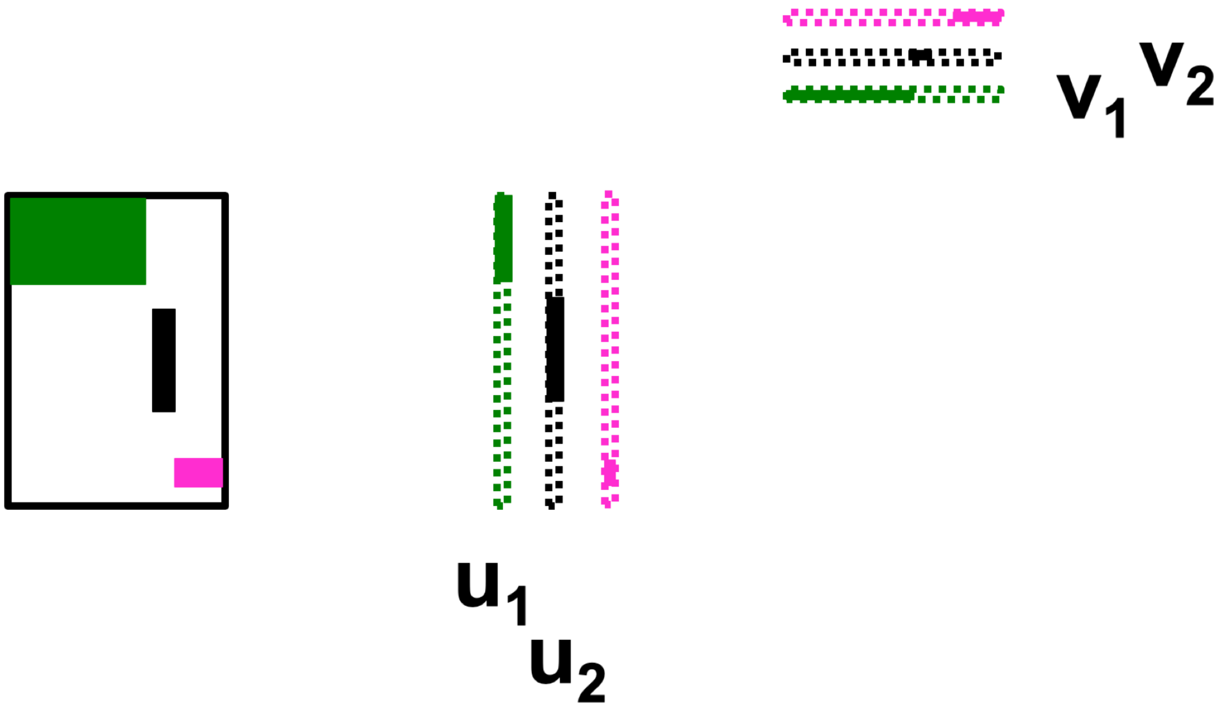
Agenda

- Motivation
- Graph Embedding
  - SVD
  - Deep Graph
- Tensor Embedding
- Knowledge Graph Embedding

Familiar embedding: SVD



Familiar embedding: SVD



SVD as embedding

- $A = \mathbf{U} \Lambda \mathbf{V}^T$

$$
\begin{array}{cc}
 & \begin{array}{ccccc}\text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung}\end{array}\\
\begin{array}{c}\text{CS}\\ \text{CS}\\ \text{CS}\\ \text{CS}\\ \text{MD}\\ \text{MD}\\ \text{MD}\end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0\\
2 & 2 & 2 & 0 & 0\\
1 & 1 & 1 & 0 & 0\\
5 & 5 & 5 & 0 & 0\\
0 & 0 & 0 & 2 & 2\\
0 & 0 & 0 & 3 & 3\\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{cc}\text{CS-concept} & \text{MD-concept}\end{array}\\
\begin{bmatrix}
0.18 & 0\\
0.36 & 0\\
0.18 & 0\\
0.90 & 0\\
0 & 0.53\\
0 & 0.80\\
0 & 0.27
\end{bmatrix}
\end{array}
\text{ x }
\begin{bmatrix}
9.64 & 0\\
0 & 5.29
\end{bmatrix}
\text{ x }
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0\\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD as embedding

$$
\begin{array}{c c}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \\ \end{array} &
\left[\begin{array}{cccc|c}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
\hline
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{array}\right]
\end{array}
=
\left[\begin{array}{cc}
0.18 & 0 \\
\boxed{0.36 \quad 0} & \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{array}\right]
\times
\left[\begin{array}{cc}
9.64 & 0 \\
0 & 5.29
\end{array}\right]
\times
\left[\begin{array}{ccccc}
0.58 & 0.58 & 0.58 & \boxed{\begin{array}{c} 0 \\ 0.71 \end{array}} & 0 \\
0 & 0 & 0 & & 0.71
\end{array}\right]
$$

$\overrightarrow{D_2}$: *embedding of document D2*

$\overrightarrow{T_4}$: *embedding of term T4*

Deep Graph Embeddings

- DeepWalk
- Node2Vec
- Metapath2Vec
- LINE
- UltimateWalk
- AutoEncoder
- Struc2Vec
- GraphSAGE
- GCN
- …

Skip-gram

Skip-gram

- Borrowed from work on language model
- Sample a set of paths with random walk from node v_i
  - $min\ -log \sum_{v_j \epsilon N(v_i)} P(v_j|v_i)$
  - $P\left(v_j \middle| v_i\right) = \frac{\exp(v_i v_j)}{\sum_{v_k \epsilon |V|} \exp(v_i v_k)}$
- Solved with
  - Hierarchical Softmax (DeepWalk)
  - Negative Sampling (Node2Vec)

Deep Graph Embeddings

- DeepWalk
- Node2Vec
- Metapath2Vec — *heterogenous graph*
- LINE — *1st order + 2nd order proximity*
- UltimateWalk — *closed form, unifies DeepWalk and Node2Vec*
- AutoEncoder — *reconstruct W, similar to SVD*
- Struc2Vec — *focuses on structural similarity*
- GraphSAGE — *"inductive", sample and aggregate*
- GCN — *interesting! borrowed the idea from CNN*
- …

Embedding can help with…

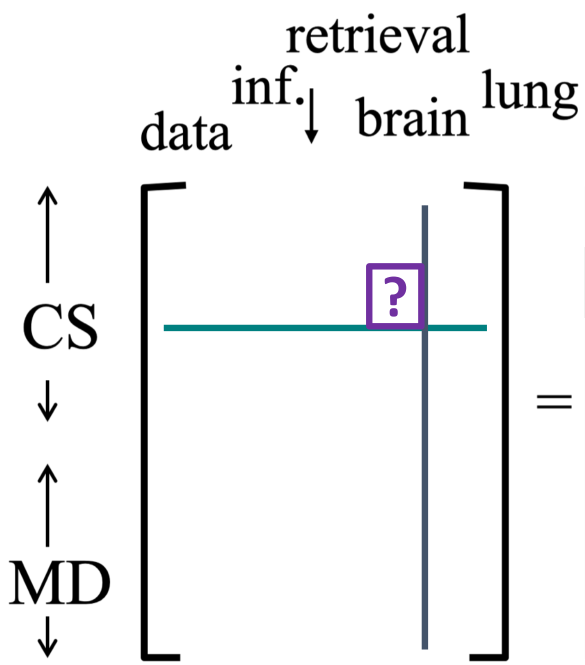
- Reconstruction / Fact checking
  - Triples completion
- Classification
  - Triples classification
- 'Featurizing'
  - (Link prediction)
  - (Recommendation)

Example: Reconstruction of (2,4)

- A: $\overrightarrow{D_2}$ x $\overrightarrow{T_4}$ = 0

$\overrightarrow{D_2}$: embedding of document D2

$\overrightarrow{T_4}$: embedding of term T4



$$= \begin{bmatrix} 0.18 & 0 \\ \boxed{0.36 \quad 0} & \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & \boxed{\begin{matrix} 0 \\ 0.71 \end{matrix}} & 0 \\ 0 & 0 & 0 & & 0.71 \end{bmatrix}$$

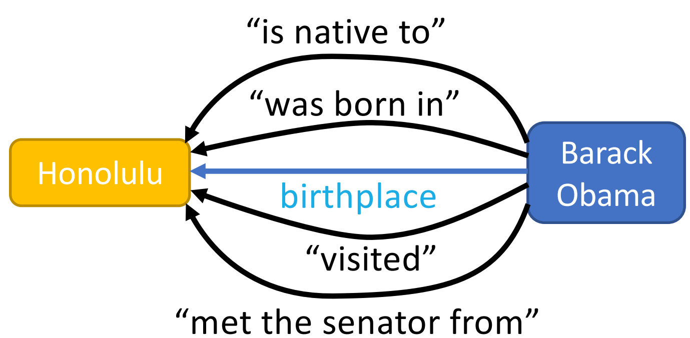
Agenda

- Motivation
- Graph Embedding
- Tensor Embedding
  - Pairs and Relations as Matrix
  - Tensor Formulation of KG
- Knowledge Graph Embedding

“Distant” Supervision

John was born in Liverpool, to Julia and Alfred Lennon.

No direct supervision gives us this information.
Supervised: Too expensive to label sentences
Rule-based: Too much variety in language
Both only work for a small set of relations, i.e. 10s, not 100s

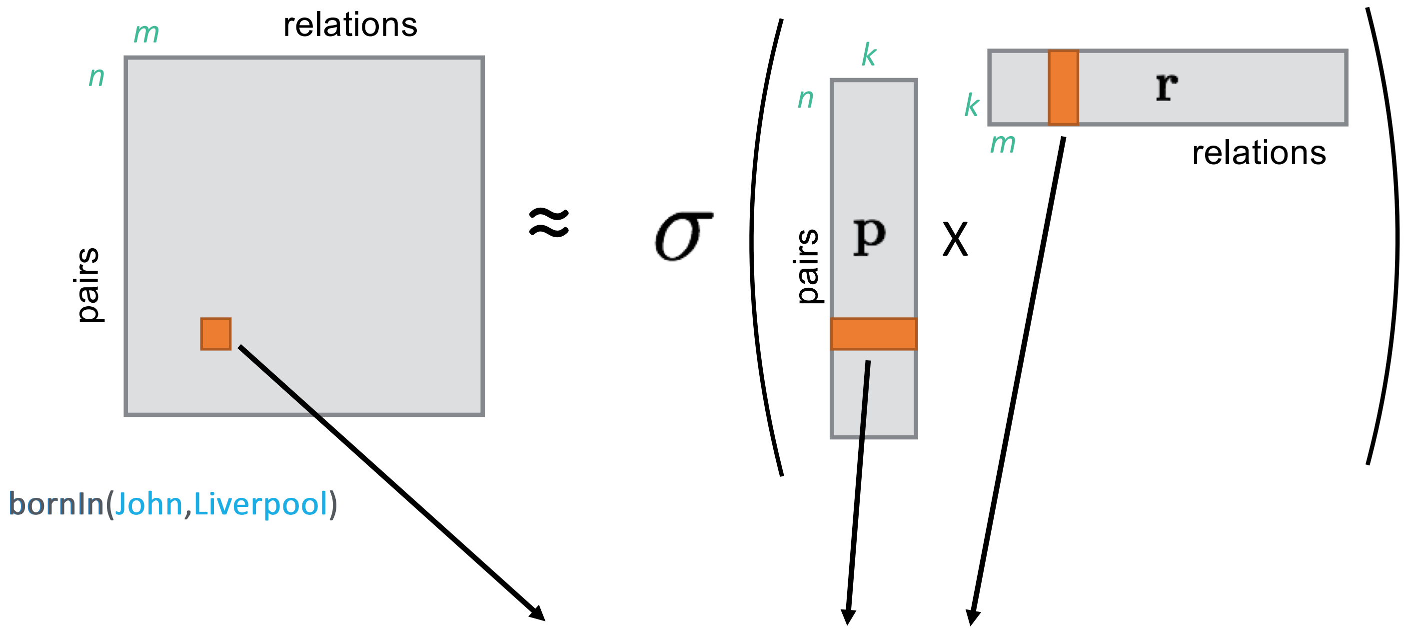


Relation Extraction as a Matrix

John was born in Liverpool, to Julia and Alfred Lennon.

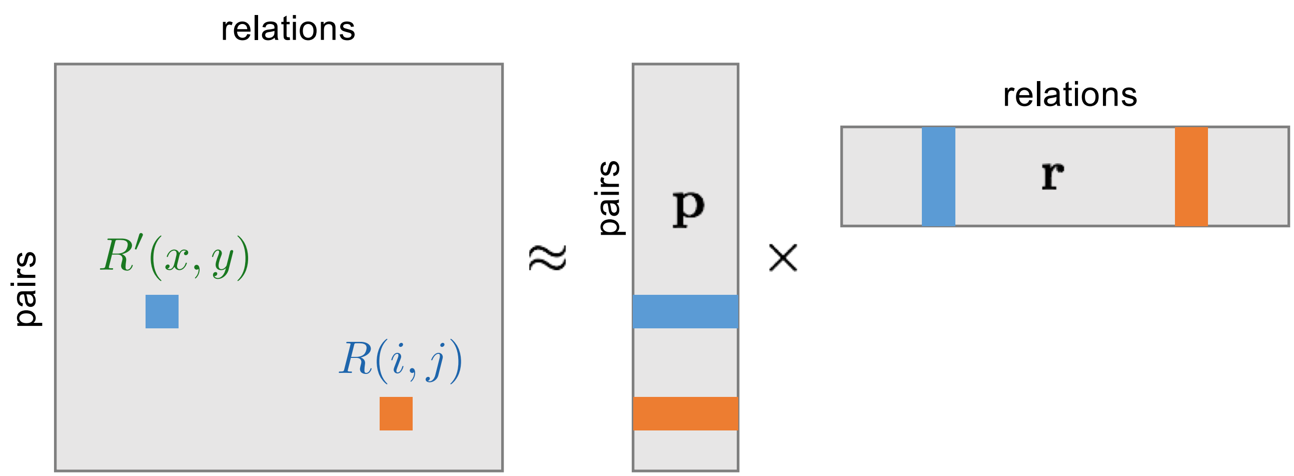
Entity Pairs	was born in <-nsubjpass-born<-nmod:in-	was born to	and	birthplace(X,Y)	spouse(X,Y)
John Lennon, Liverpool	1			?	
John Lennon, Julia Lennon		1			
John Lennon, Alfred Lennon		1			
Julia Lennon, Alfred Lennon			1		?
Barack Obama, Hawaii	1			1	
Barack Obama, Michelle Obama			1		1

Matrix Factorization



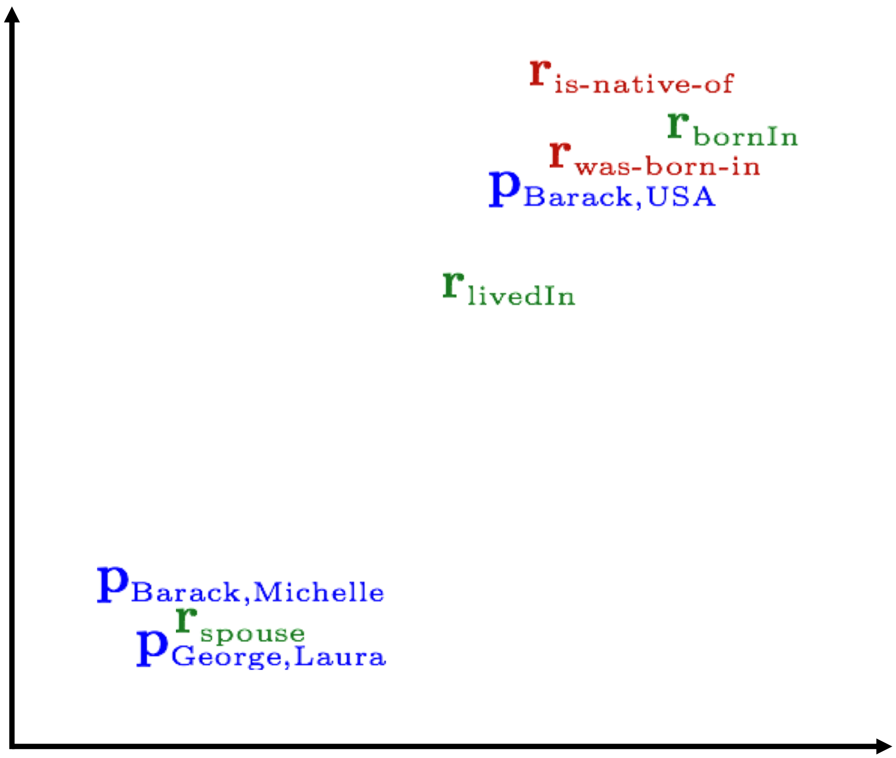
$$P(R(i,j)) = \sigma(\mathbf{p}_{i,j} \cdot \mathbf{r}_R)$$

Training: Stochastic Updates



- Pick an observed cell, $R(i,j)$:
  - Update $\mathbf{p}_{ij}$ & $\mathbf{r}_R$ such that $R(i,j)$ is higher
- Pick any random cell, assume it is negative:
  - Update $\mathbf{p}_{xy}$ & $\mathbf{r}_{R'}$ such that $R'(x,y)$ is lower

Relation Embeddings



Embeddings ~ Logical Relations

Relation Embeddings, r

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

Entity Pair Embeddings, p

- 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

(Freeman,Harvard)
→ (Boyle,OhioState)

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

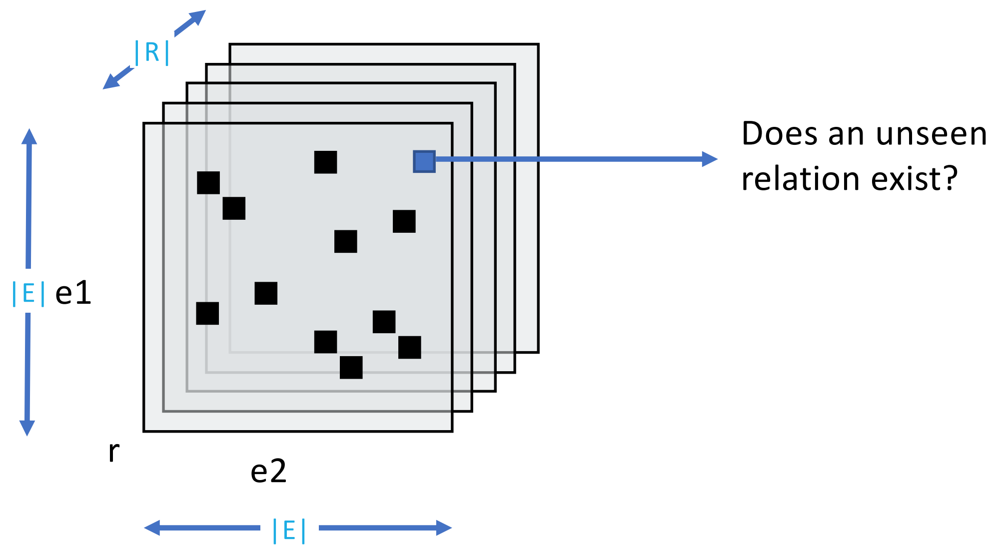
Learns asymmetric entailment:

PER historian at UNIV $\rightarrow$ PER professor at UNIV

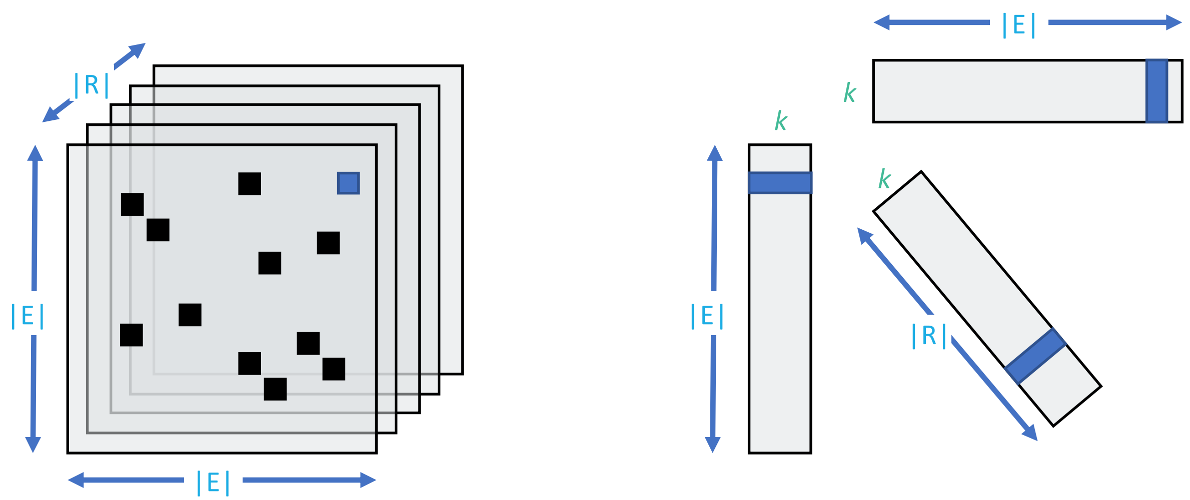
But,

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

Tensor Formulation of KG



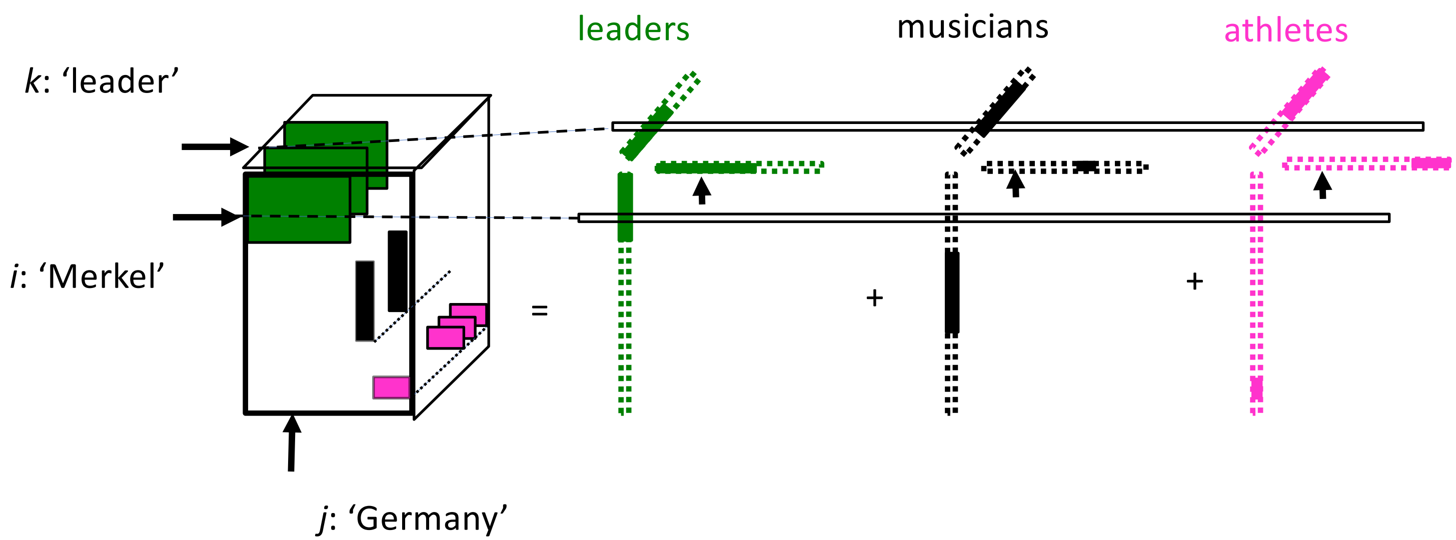
Factorize that Tensor



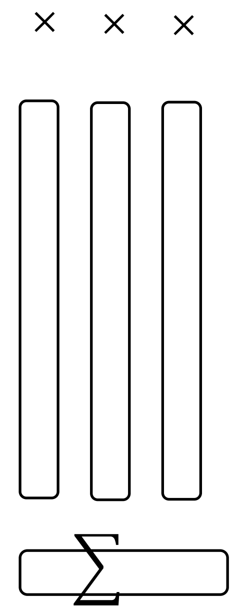
$$S(r(a,b)) = f(\mathbf{v}_r, \mathbf{v}_a, \mathbf{v}_b)$$

PARAFAC: as embedding

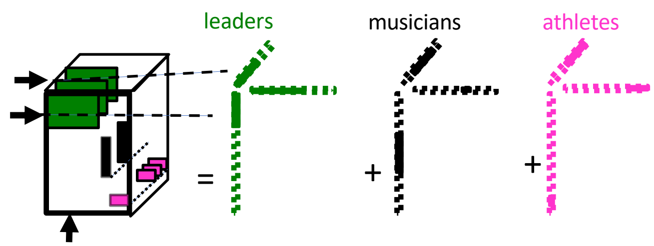
- ‘Merkel’: i-th subject vector: (1,0,0)
- ‘Germany’: j-th object vector: (1,0,0)
- ‘is_leader’: k-th verb vector: (1,0,0)



Reconstruction



- ‘Merkel’: i-th subject vector: $\vec{s}$=(1, 0, 0)
- ‘Germany’: j-th object vector: $\vec{o}$=(1, 0, 0)
- ‘is_leader’: k-th verb vector: $\vec{v}$=(1, 0, 0)
- A: $x_{i,j,k} = \sum_{h=1}^{3} s_{i,h}\, o_{j,h}\, v_{k,h}$
- Intuitively:
  - s,v,o: should have common ’concepts’



Agenda

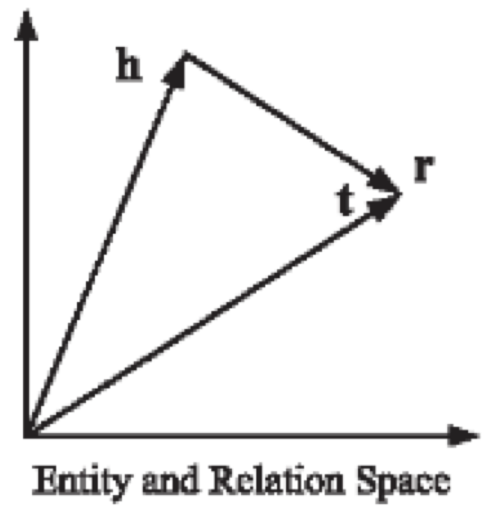
- Motivation
- Graph Embedding
- Tensor Embedding
- Knowledge Graph Embedding
    - Triple Scoring
        - Addition
        - Multiplication
    - Loss
    - Applications

Knowledge Graph Embedding

- **Triple scoring**: what is the relationship among sub (h), pred (r), and obj (t)?
  - Addition: h + r =?= t
  - Multiplication: h ∘ r =?= t
- **Loss**: what shall we optimize?
  - Closed-world assumption
  - Open-world assumption

Triple Scoring - Addition

- Addition: $\mathbf{h} + \mathbf{r} =?= \mathbf{t}$
  - TransE
    - score(h,r,t) = $- ||\mathbf{h}+\mathbf{r}-\mathbf{t}||_{1/2}$



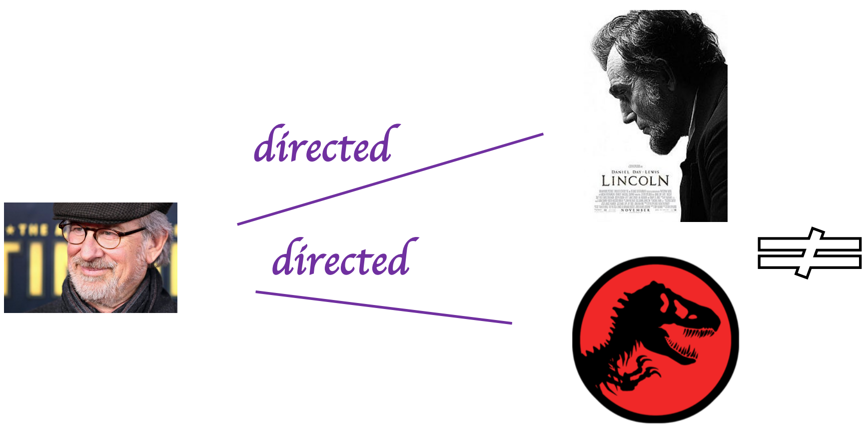
TransE

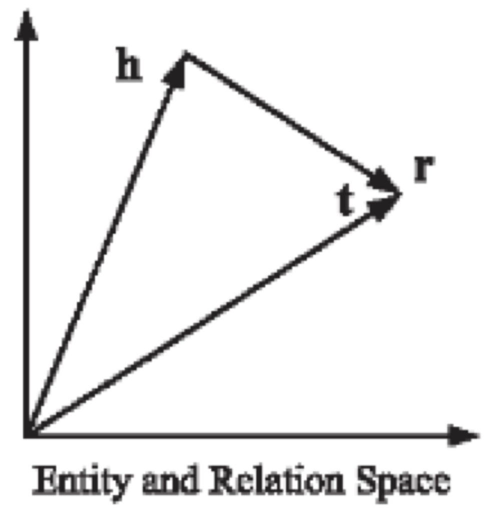
‘Merkel’: $\vec{h}$=(1, 0, 0)
‘Germany’: $\vec{t}$=(1, 1, 0)
‘is_leader’: $\vec{r}$=(0, 1, 0)
$score\,(\mathrm{h, r, t}) = -\,||\, \vec{h} + \vec{r} - \vec{t}\,||_{1/2} = 0$

‘Merkel’: $\vec{h}$=(1, 0, 0)
‘Beatles’: $\vec{t'}$=(0, 0, 1)
‘plays_bass’: $\vec{r'}$=(0, 0, 1)
$score\,(\mathrm{h, r, t}) = -\,||\, \vec{h} + \vec{r'} - \vec{t'}\,||_{1/2} = -1$

Triple Scoring - Addition

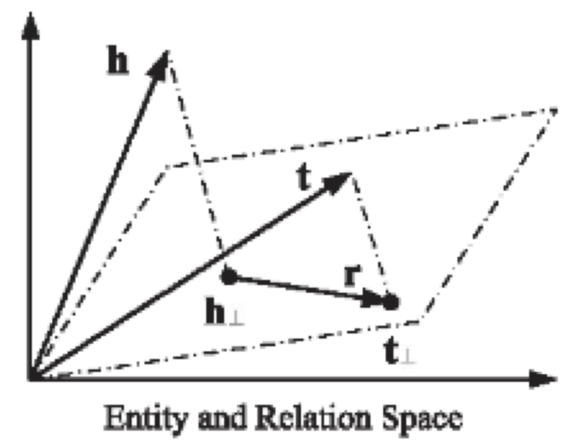
- Addition: $\mathbf{h} + \mathbf{r} =?= \mathbf{t}$
  - TransE
    - score(h,r,t) = $-\ ||\mathbf{h}+\mathbf{r}-\mathbf{t}||_{1/2}$
  - What if multiple objects apply??





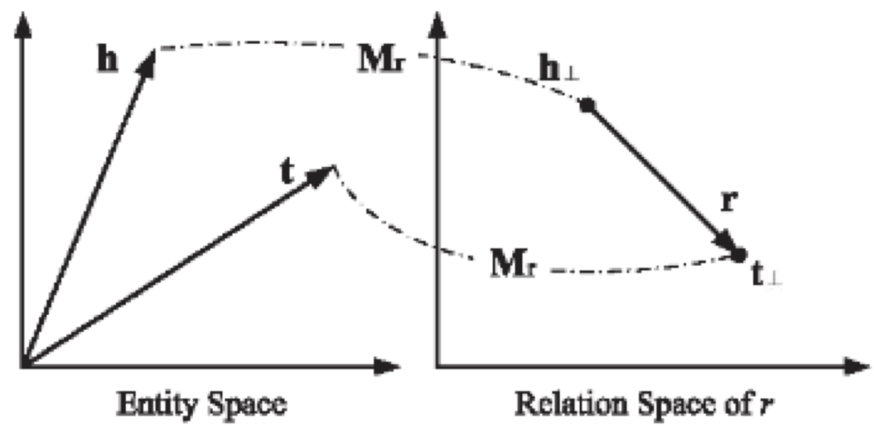
Triple Scoring - Addition

- Addition: $\mathbf{h} + \mathbf{r}$ =?= $\mathbf{t}$
  - TransE
    - score(h,r,t) = $-\ ||\mathbf{h}+\mathbf{r}-\mathbf{t}||_{1/2}$
  - TransH
    - project to relation-specific hyperplanes



Triple Scoring - Addition

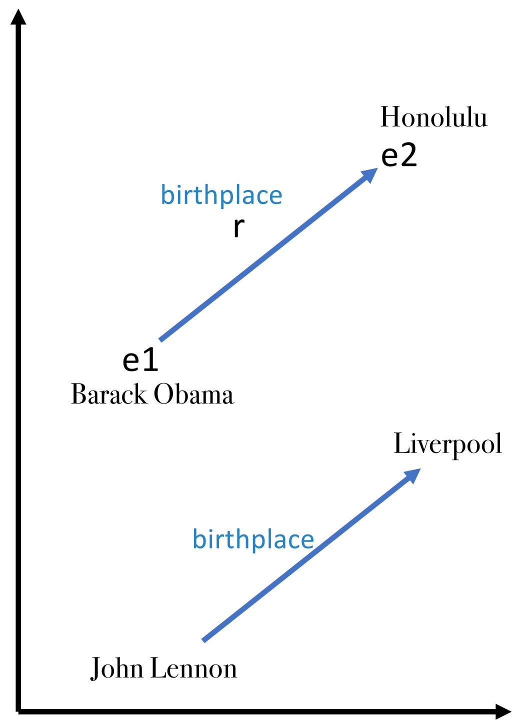
- Addition: **h** + **r** =?= **t**
  - TransE
    - score(h,r,t) = – ||**h**+**r**-**t**|| $_{1/2}$
  - TransH
    - project to relation-specific hyperplanes
  - TransR
    - translate to relation-specific space



Triple Scoring - Addition

- Addition: $\mathbf{h} + \mathbf{r} =?= \mathbf{t}$
  - TransE
    - score(h,r,t) = $-\,||\mathbf{h}+\mathbf{r}-\mathbf{t}||_{1/2}$
  - TransH
    - project to relation-specific hyperplanes
  - TransR
    - translate to relation-specific space
  - Many simplifications of TransH and TransR
    - STransE is reported to be the best in
      Dat Quoc Nguyen. An overview of embedding models of entities and relationships for knowledge base completion

Triple Scoring - Addition



TransE

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

TransH

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

$$\mathbf{e}_a^{\perp} = \mathbf{e}_a - \mathbf{w}_r^T \mathbf{e}_a \mathbf{w}_r$$

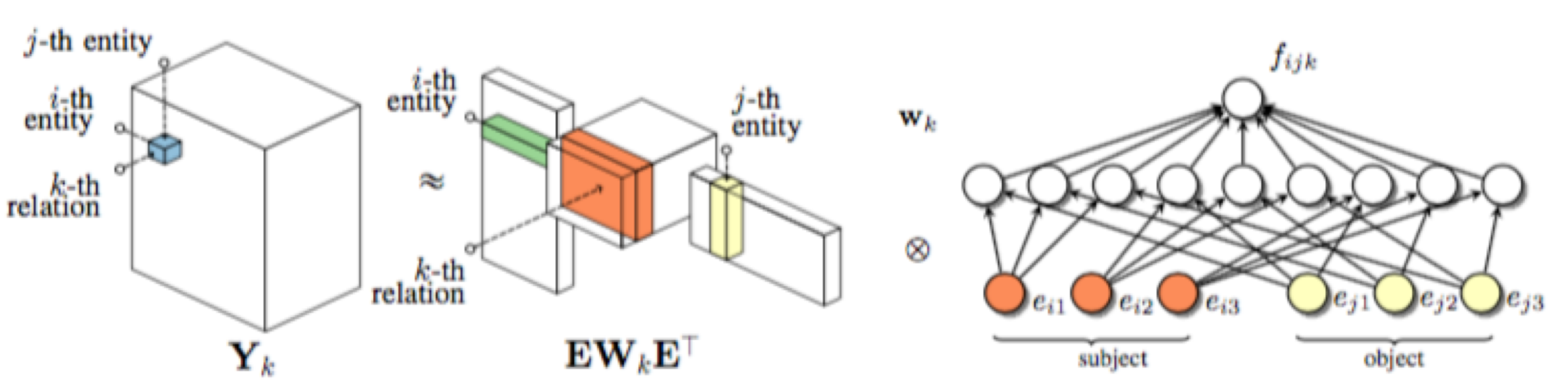
TransR

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

Triple Scoring - Multiplication

- Multiplication: h ∘ r =?= t
  - RESCAL: score(h,r,t) = $\mathbf{h}^{\top}\mathbf{W}_r\mathbf{t}$

    Too many parameters?!



Triple Scoring - Multiplication

- Multiplication: h ∘ r =?= t
  - RESCAL: score(h,r,t) = $\mathbf{h}^\top \mathbf{W}_r \mathbf{t}$
  - DistMult: score(h,r,t) = $\mathbf{h}^\top \mathrm{diag}(\mathbf{r})\mathbf{t}$
    Simplify RESCAL by using a diagonal matrix

RESCAL

'Merkel': $h=(1, 0)^{T}$

'Germany': $t=(0, 1)^{T}$

'is_leader': $W_r=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$

score $(h, r, t) = h^{\top}W_r t$

$= \sum(h \otimes t) \odot W_r = 1$

DistMult

'Merkel': $h=(1, 0)^{T}$

'Germany': $t=(1, 0)^{T}$

'is_leader': $W_r=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$

score $(h, r, t) = h^{\top}W_r t$

$= \sum(h \odot t) \odot \mathrm{diag}(W_r) = 1$

Triple Scoring - Multiplication

- Multiplication: h ∘ r =?= t
  - RESCAL: score(h,r,t) = $\mathbf{h}^\top \mathbf{W}_r \mathbf{t}$
  - DistMult: score(h,r,t) = $\mathbf{h}^\top \mathrm{diag}(\mathbf{r})\mathbf{t}$
    Simplify RESCAL by using a diagonal matrix
    - Cannot deal with asymmetric relations!!
  - ComplEx: score(h,r,t) = $\mathrm{Re}(\mathbf{h}^\top \mathrm{diag}(\mathbf{r})\mathbf{t})$
    Extend DistMult by introducing complex value embedding,
    so can handle asymmetric relations

ComplEx

- $h = R(h) + iI(h), \quad t = R(t) + iI(t), \quad r = R(r) + iI(r)$

- $$\begin{aligned} h \odot \bar{t} &= \big(R(h) + iI(h)\big) \odot \big(R(t) + iI(t)\big) \\ &= R(h) \odot R(t) + I(h) \odot I(t) \\ &\quad +i(I(h) \odot R(t) - R(h) \odot I(t)) \end{aligned}$$

- $$\begin{aligned} Re\{\big(h \odot \bar{t}\big) \odot r\} &= R(h) \odot R(t) \odot R(r) \\ &\quad +I(h) \odot I(t) \odot R(r) \\ &\quad +R(h) \odot I(t) \odot I(r) \\ &\quad -I(h) \odot R(t) \odot I(r) \end{aligned}$$

ComplEx

- $score(h,r,t) = \sum Re\{(h \odot \bar{t}) \odot r\}$

$$
\begin{aligned}
&= \sum R(h) \odot R(t) \odot R(r) \quad \Rightarrow \quad \text{DistMult} \\
&\left.\begin{aligned}
&+\sum I(h) \odot I(t) \odot \underline{R(r)} \\
&+\sum \underline{R(h)} \odot I(t) \odot I(r) \\
&-\sum I(h) \odot \underline{R(t)} \odot I(r)
\end{aligned}\right]
\end{aligned}
$$

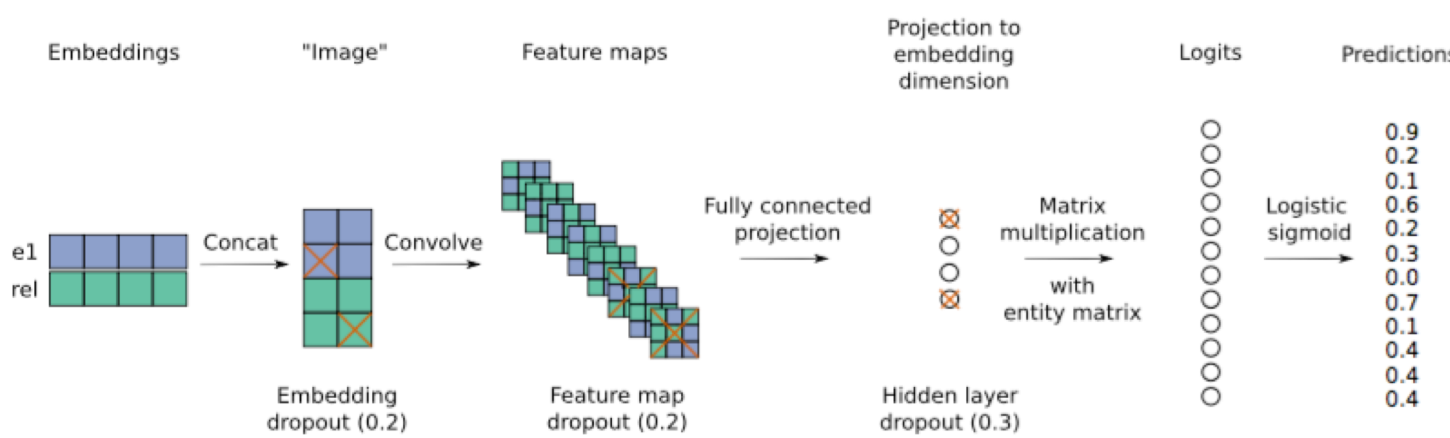
- $\neq score(t,r,h)$ $\quad \Rightarrow \quad$ Asymmetry

Triple Scoring - Multiplication

- Multiplication: h ∘ r =?= t
  - RESCAL: $\text{score}(h,r,t) = \mathbf{h}^\top \mathbf{W}_r \mathbf{t}$
  - DistMult: $\text{score}(h,r,t) = \mathbf{h}^\top \text{diag}(\mathbf{r})\mathbf{t}$
  - ComplEx: $\text{score}(h,r,t) = \text{Re}(\mathbf{h}^\top \text{diag}(\mathbf{r})\mathbf{t})$
  - ConvE: Use convolutional NN to reduce parameters

ComplEx and ConvE have state-of-the-art results
- Reduce parameters
- Certain flexibility

DistMult is light-weight, and good in practice.



Loss

- Closed world assumption: square loss

$$L = \sum_{h,t \in E, r \in R} (y_{h,r,t} - f(h,r,t))^2$$

- Open world assumption: triplet loss

$$L = \sum_{T+} \sum_{T-} max(0, \gamma - f(h,r,t) + f(h',r',t'))$$

OWA works best

KGE Applications

- Learn embeddings from IMDb data and identify WikiData errors, using DistMult

Subject	Relation	Target	Reason
The Moises Padilla Story	writtenBy	César Ámigo Aguilar	Linkage error
Bajrangi Bhaijaan	writtenBy	Yo Yo Honey Singh	Wrong relationship
Piste noire	writtenBy	Jalil Naciri	Wrong relationship
Enter the Ninja	musicComposedBy	Michael Lewis	Linkage error
The Secret Life of Words	musicComposedBy	Hal Hartley	Cannot confirm
…	…	…	…

Comparing Real KGs with Benchmarks

- Examine statistics of real KGs and derived benchmarks
- Two metrics for capturing data distribution and sparsity:
  - entity & relation entropy (EE/RE) – measure diversity of facts
  - entity & relation density (ED/RD) – concentration of facts

	KG	Triples	Entities	Rels	EE	RE	ED	RD	Prec
Real	Freebase	1B	124M	15K	14	3.2	16	68K	1
	NELL1000	92M	4.8M	435	21	4.9	19	210K	0.45
	WordNet	380K	116K	27	21	2.3	7	21K	1
Bench.	FB15K	592K	15K	1.3K	16	5.1	79	440	1
	NELL165	1M	820K	221	25	1.5	3	4.7K	0.35
	WN18	151K	40K	18	19	2.1	7	8.4K	1

Comparing Real KGs with Benchmarks

	KG	Triples	Entities	Rels	EE	RE	ED	RD	Prec
Real	Freebase	1B	124M	15K	14	3.2	16	68K	1
	NELL1000	92M	4.8M	435	21	4.9	19	210K	0.45
	WordNet	380K	116K	27	21	2.3	7	21K	1
Bench.	FB15K	592K	15K	1.3K	16	5.1	79	440	1
	NELL165	1M	820K	221	25	1.5	3	4.7K	0.35
	WN18	151K	40K	18	19	2.1	7	8.4K	1

Observations:

- Freebase is largest KG with highest RD, but lowest EE
- NELL1000 is diverse (high EE/RE), highest RD, low precision
- WN/WN18 are much smaller, low rels, low RE, low ED
- FB15K has very high ED, very low RD, more diverse than FB
- NELL165 has lowest ED, highest EE, lowest RE, low precision

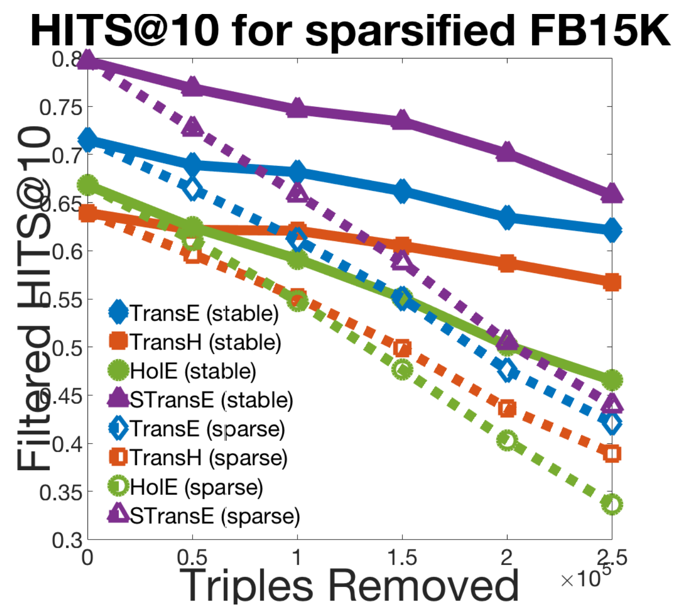
Do embeddings work for extracted KGs?

- **Approach:**
  - Evaluate on the NELL knowledge graph, containing millions of candidates extracted from WWW text
- **Observations:**
  - Baseline (threshold input) wins against embeddings
  - Best results from graphical model (PSL-KGI) using rules & uncertainty
  - More complex embedding methods have the worst performance
- **Conclusion:**
  - Embeddings have poor performance on sparse & noisy KGs extracted from text

Method	AUC	F1
TransH	0.701	0.783
HolE	0.710	0.783
TransE	0.726	0.783
STransE	0.784	0.783
Baseline	0.873	0.828
PSL-KGI	**0.891**	**0.848**

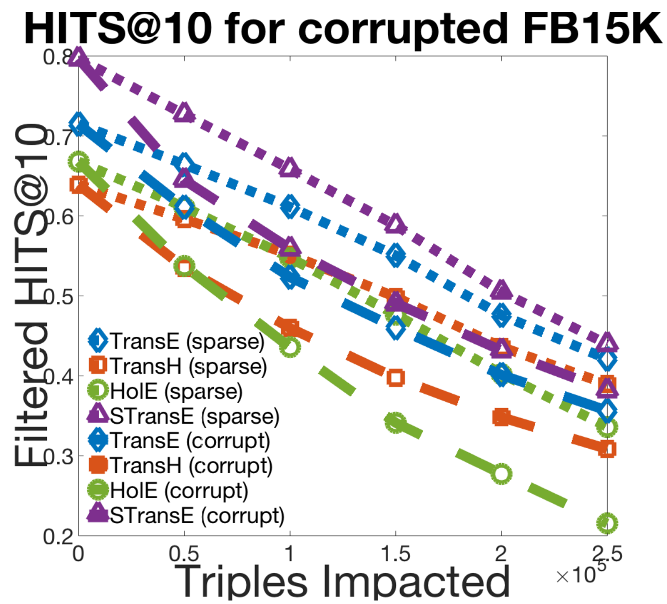
Do embeddings require complete KGs?

- **Approach:**
  - Remove training data, either in clusters to maintain relation density (stable) or randomly (sparse)
- **Observations:**
  - All methods perform much worse with sparse KGs relative to stable baseline
  - At 50% removal, stable can outperform sparse by 60%
  - STransE most sensitive, HolE least sensitive to sparsity
- **Conclusion:**
  - performance quickly degrades with sparsity



Do embeddings require reliable KGs?

- **Approach:**
  - Randomly "`corrupt`" training data by altering subject, predicate, or object
- **Observations:**
  - `corrupt` training data is worse than `sparse` data
  - Deficit between sparse and `corrupt` remains stable
  - HolE most sensitive, STransE least sensitive to corruption
- **Conclusion:**
  - Unreliable data harms training more than missing data



When is noisy data worth using?

- **Approach:**
  - Start with sparse training set and add new training data with differing noise levels
- **Observations:**
  - All methods receive boost from initial noisy data
  - Enough low noise data can allow recovery
  - Even very noisy data doesn't degrade performance much
- **Conclusion:**
  - Extending sparse training data with noisy inputs can help performance