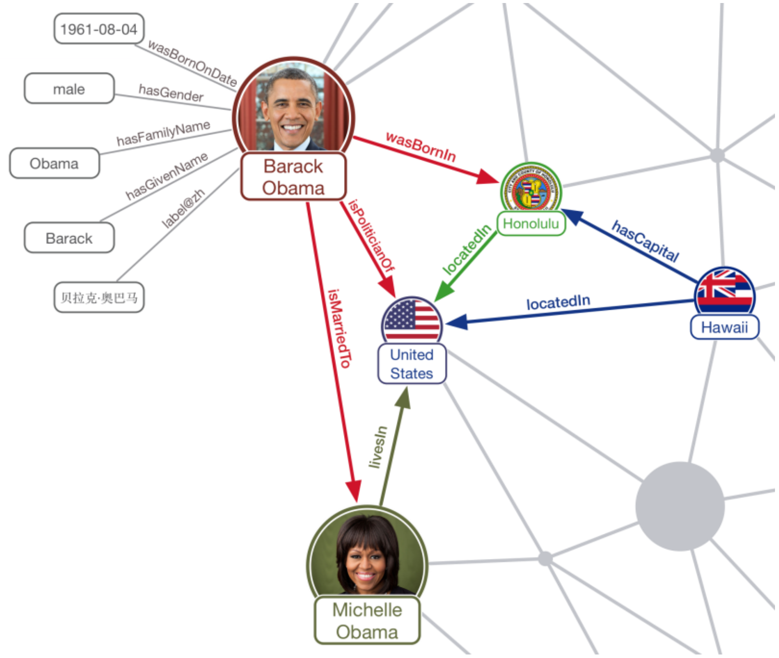
Neural Link Prediction for Multi-Modal Knowledge Graphs

Mathias Niepert and Alberto Garcia-Duran

NEC Labs Europe
Heidelberg



Outline

- Quick Reminder
- Failing with Latent and Relational Models
- Simple Link Prediction in KGs:
  - Graph Structure and Numerical Information
  - Visual Information
- Simple Link Prediction in Temporal KGs
- Remarks

Outline

- **Quick Reminder**

- Failing with Latent and Relational Models

- Simple Link Prediction in KGs:
  - Graph Structure and Numerical Information
  - Visual Information

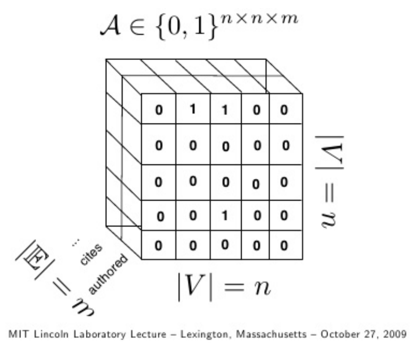
- Simple Link Prediction in Temporal KGs

- Remarks

Quick Reminder

Tensor Factorization problem

- One adjacency matrix per relationship
- This is not new!



Latent Models

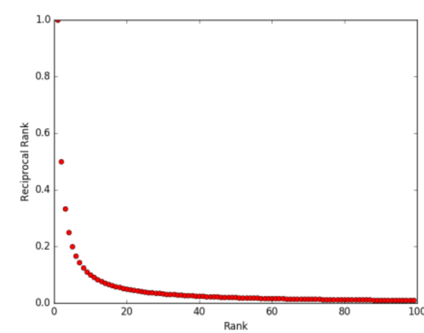
- Scoring function operating on latent space
- At a high level, relationships can be seen as operations/transformations operating on the entities
- Parameter sharing between head and tail arguments

Relational Models

- Extraction of relational features (e.g. via rule miners such as AMIE+)
- Scoring function operating on relational features

Evaluation Metrics

- Queries of the type (h, r, ?) or (?, r, t)
- MRR is the most informative evaluation metric

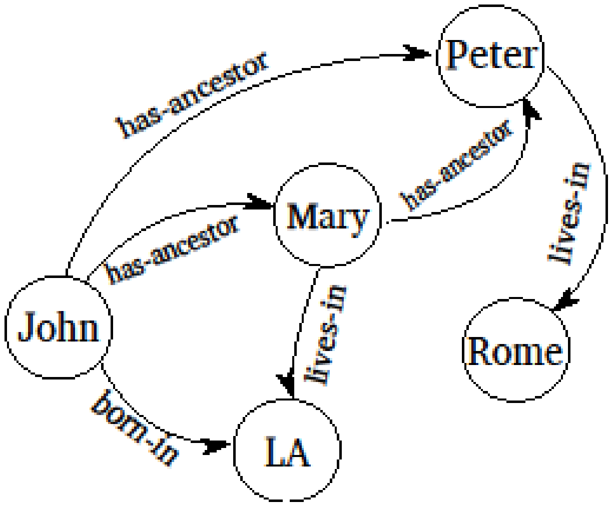


Outline

- Quick Reminder
- **Failing with Latent and Relational Models**
- Simple Link Prediction in KGs:
  - Graph Structure and Numerical Information
  - Visual Information
- Simple Link Prediction in Temporal KGs
- Remarks

"Compositionality" [Guu et al.,2016]

- They perform random walks in the KG and recursively apply the same transformation



Perform two random walks in the graph:

(John, has-ancestor, lives-in, LA)
(John, born-in, LA)

Failing with Latent and Relational Models
Latent Models

- Apply the same transformation recursively (e.g. TransE):

John + ancestor + lives-in ≈ LA

John + born-in ≈ LA

- If this happens this very often, then:

ancestor + lives-in ≈ born-in

- This amounts to learn the horn rule:

(h, has-ancestor, x) ^ (x, lives-in, t) -> (h, born-in, t)

has-ancestor ^ lives-in -> born-in

- Let's perform two more random walks:

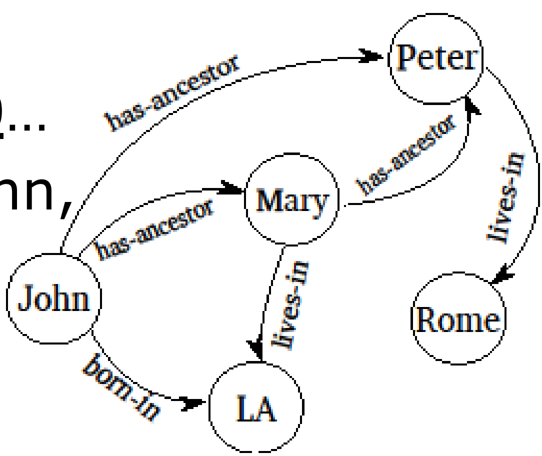
John + ancestor + ancestor ≈ Peter

John + ancestor ≈ Peter

- This happens this very often, then:

ancestor + ancestor ≈ ancestor

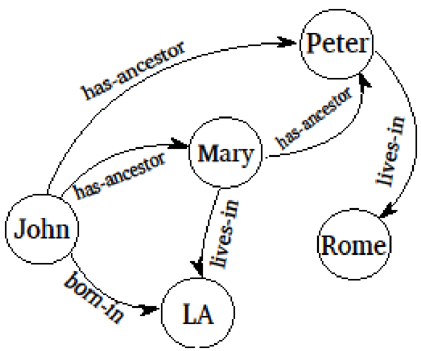
- Only possible if the embedding for ancestor is **0**…
- … but this would collapse the embeddings of John, Mary and Peter to the same point. Not good…



For relational models to learn the predictive power of paths is easy.

- As long as we have enough examples of each path

But the number of possible paths grows exponentially with the number of relationships and length of the path



has-ancestor ^ has-ancestor

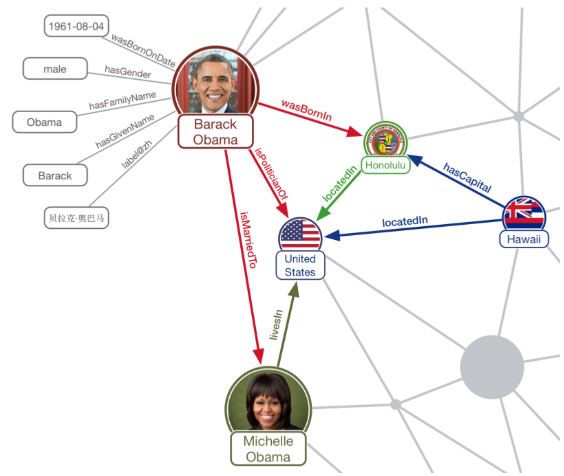
has-ancestor ^ has-ancestor ^ lives-in

has-ancestor ^ lives-in

$(\text{born-in})^{-1}$ ^ has-ancestor

$(\text{born-in})^{-1}$ ^ has-ancestor ^ has-ancestor

Try this



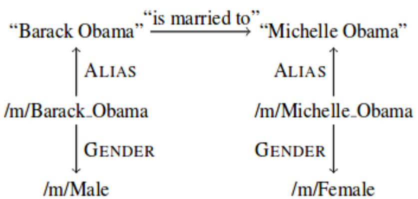
Failing with Latent and Relational Models
Relational Models

We cannot mine all possible paths in a graph

- No paths, no party!

[Neelankatan et al.,2015; Gardner et al.,2015] used relational features other than paths

- Path-bigram features
- One-sided features



Subgraph for /m/Barack_Obama	
π	i
-ALIAS-	"Barack Obama"
-GENDER-	/m/Male
-ALIAS-"is married to"-	"Michelle Obama"

Subgraph for /m/Michelle_Obama	
π	i
-ALIAS-	"Michelle Obama"
-GENDER-	/m/Female
-ALIAS-"is married to"$^{-1}$-	"Barack Obama"

For medium/large KGs the space of possible relational features is huge

- AMIE fails to obtain rules whose bodies are of up to 2 atoms in the Decagon data set (20k entities and 1k relation types) [Zitnik et al., 2018]

- Latent methods learn (at least) entity type information

- Latent methods very helpful for sparse KGs

- Latent models fail at learning very simple horn rules

- Dichotomy of relational features: either work perfectly or fail completely (random)

- Relational features outperform embedding methods on KBs with dense relational structure

		Rank	
Query	Correct entity	TE	GM
nationality(?, US)	W. H. Macy	2	233
born_here(HK, ?)	W. Chau-sang	5	135
contains(?, Curtis-Inst)	USA	32	1
children(?, H. Roshan)	R. Roshan	26	1

Failing with Latent and Relational Models

- Can we use latent models in conjunction with "simple" relational features?
- Acknowledgement: There are a number of recent approaches [Rocktaschel et al.,2015; Guo et al.,2016; Minervini et al.,2017] that combine relational and latent representations by explicitly incorporating known logical rules into the embedding learning formulation
- [Guu et al.,2016] implicitly learns these logical rules! But we have learned that latent models have problem to learn a simple rule like:

has-ancestor ^ has-ancestor -> has-ancestor

- In general, they have problems to learn rules wherein the relationship of the head of the rule also appears in the body:

rel_1 ^ rel_2 -> rel_1

rel_1 + rel_2 ≠ rel_1 (TransE)

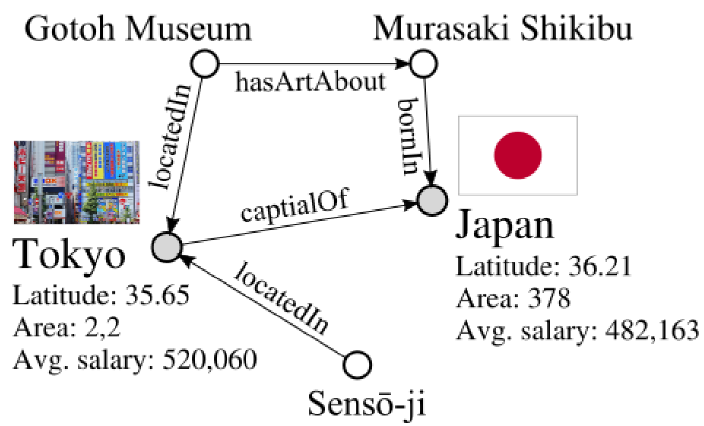
rel_1*rel_2 ≠ rel_1 (distMult, RESCAL)

Outline

- Quick Reminder
- Failing with Latent and Relational Models
- **Simple Link Prediction in KGs:**
  - **Graph Structure and Numerical Information**
  - Visual Information
- Simple Link Prediction in Temporal KGs
- Remarks

Simple Link Prediction in KGs

- Learn from all available information in the knowledge graph

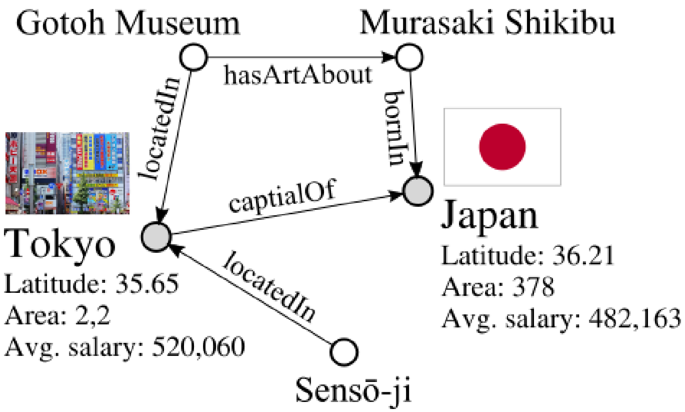


- Latent and Relational models only use graph structure information

- [Wang et al.,2016; An et al.,2018] exploit textual information of entities and relationships

- What about other data modalities?

- KG is given as a set of observed triples of the form (h, r, t)



- We aim to combine an arbitrary number of feature types F

- **KBlrn** [Garcia-Duran et al.,2017a]**:** It is a product of experts approach wherein one expert is trained for each (relation type r, feature type F) pair
  - We focus on **l**atent, **r**elational and **n**umerical features
  - Generally, we may have more than an expert for the same feature type

- Product of Experts [Hinton,2000]:

$$p(\mathbf{d}|\theta_1 \ldots \theta_n) = \frac{\Pi_m p_m(\mathbf{d}|\theta_m)}{\sum_{\mathbf{c}} \Pi_m p_m(\mathbf{c}|\theta_m)}$$

where **c** indexes all possible vectors in the data space

- From [Hinton,2000]:

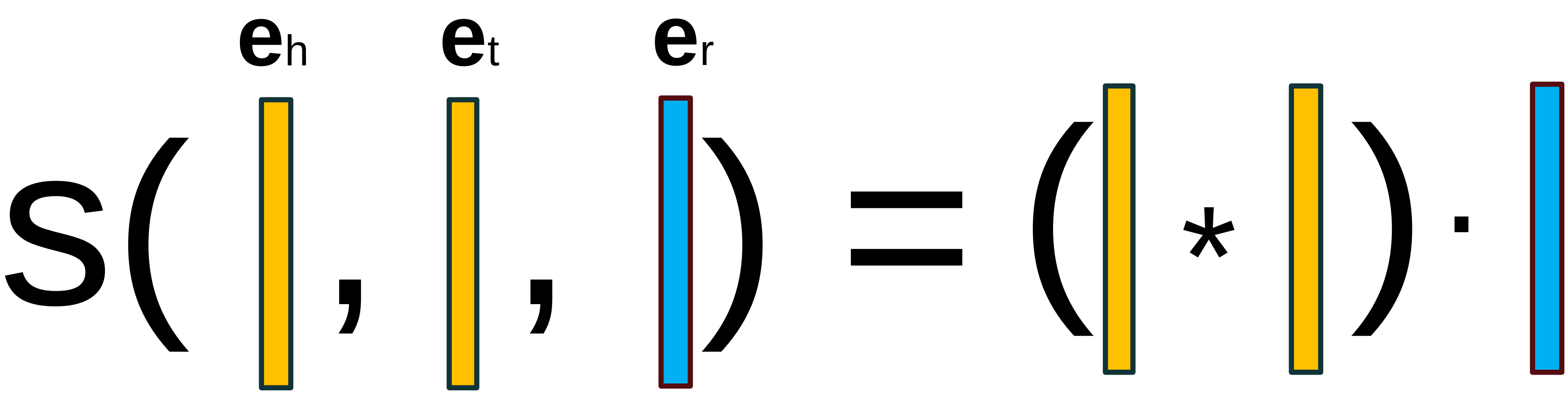
"… so long as p_m is positive it does not need to be a probability at all …"

Simple Link Prediction in KGs

KBlrn: Latent Expert

- We pick a latent model (e.g. distMult)

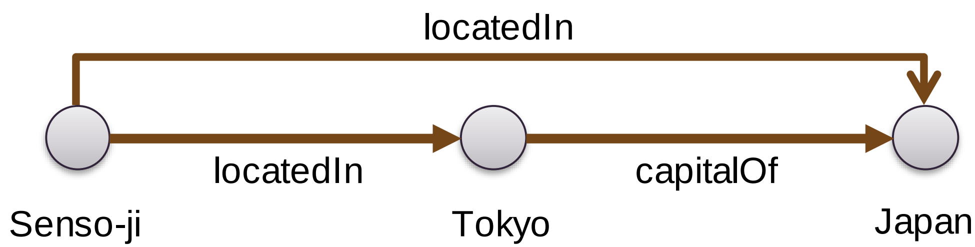
$$\mathrm{d} = (h, r, t)$$



- and force its output to be positive

$$f_{(\mathtt{r},\mathtt{L})}(\mathbf{d} \mid \theta_{(\mathtt{r},\mathtt{L})}) = \exp((\mathbf{e}_{\mathtt{h}} * \mathbf{e}_{\mathtt{t}}) \cdot \mathbf{w}^{\mathtt{r}})$$

- Horn rules whose bodies have up to 2 atoms



- We use **AMIE+** for the mining of closed horn rules

Body of the Rule (Path)	Head of the Rule
$\exists$x (h, containedby, x) $\wedge$ (t, locations_in_this_time_zone, x)	(h, time_zone, t)
$\exists$x (t, prequel, x) $\wedge$ (x, character, h)	(h, character_in_film, t)
$\exists$x (x, cause_of_death, h) $\wedge$ (x, cause_of_death, t)	(h, includes_causes_of_death, t)

- "Bag-of-paths" for each relationship

- Relational expert: $\mathrm{d} = (h, r, t)$

$$f_{(\mathtt{r},\mathtt{R})}(\mathbf{d} \mid \theta_{(\mathtt{r},\mathtt{R})}) = \exp\left(\mathbf{r}_{(\mathtt{h},\mathtt{t})} \cdot \mathbf{w}^{\mathtt{r}}_{\mathtt{rel}}\right)$$

Product of Experts:

$$p(\mathbf{d}|\theta_1 \ldots \theta_n) = \frac{\Pi_m p_m(\mathbf{d}|\theta_m)}{\sum_{\mathbf{c}} \Pi_m p_m(\mathbf{c}|\theta_m)}$$

KBlr:

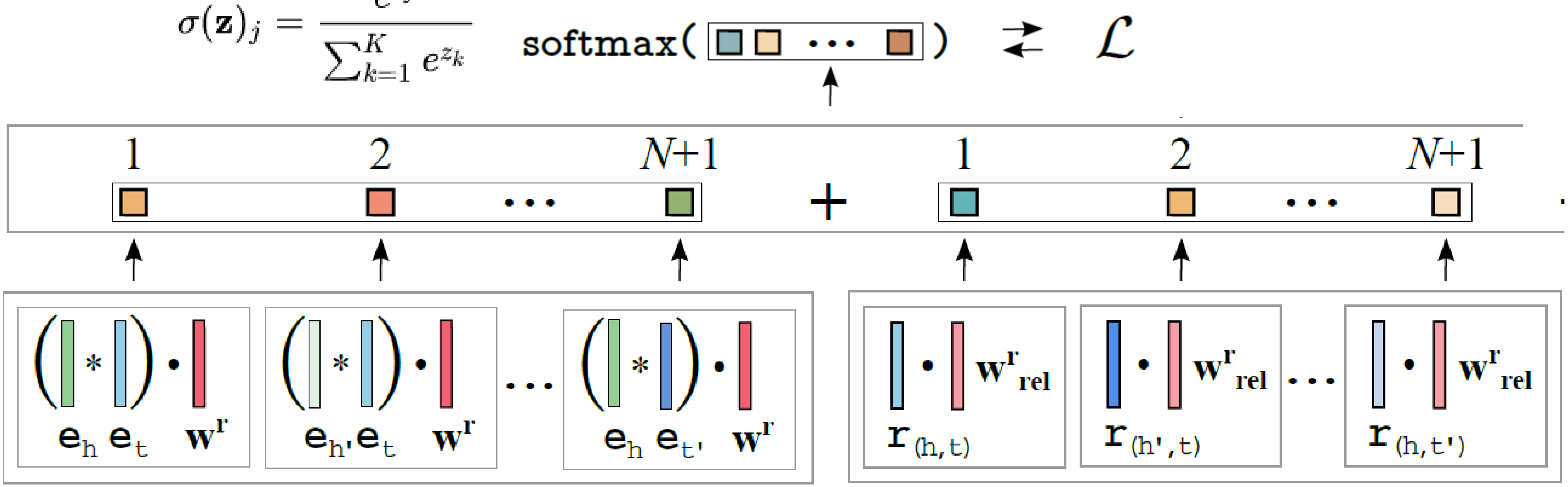
$$p(\mathtt{d} \mid \theta_1, \ldots, \theta_n) = \frac{\prod_{\mathtt{F} \in \{\mathtt{R},\mathtt{L}\}} f_{(\mathtt{r},\mathtt{F})}(\mathtt{d} \mid \theta_{(\mathtt{r},\mathtt{F})})}{\sum_{\mathtt{c}} \prod_{\mathtt{F} \in \{\mathtt{R},\mathtt{L}\}} f_{(\mathtt{r},\mathtt{F})}(\mathtt{c} \mid \theta_{(\mathtt{r},\mathtt{F})}))}$$

In practice this amounts to…

d = (*h*, *r*, *t*)

1. Sample N negative triples per positive triple
2. Compute scores of embedding and relational model for each
3. Sum scores and apply *softmax* function
4. Apply categorical cross-entropy loss

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$



KBlrn: Performance

- Competitive with more complex KB completion models
- FB15k, FB15k-237, FB122, WN18

Metrics:

MR: Mean rank of correct triple

MRR: Mean reciprocal rank

Hits@1: Percentage of correct triples ranked 1

Hits@10: Percentage of correct triples ranked in the tope 10

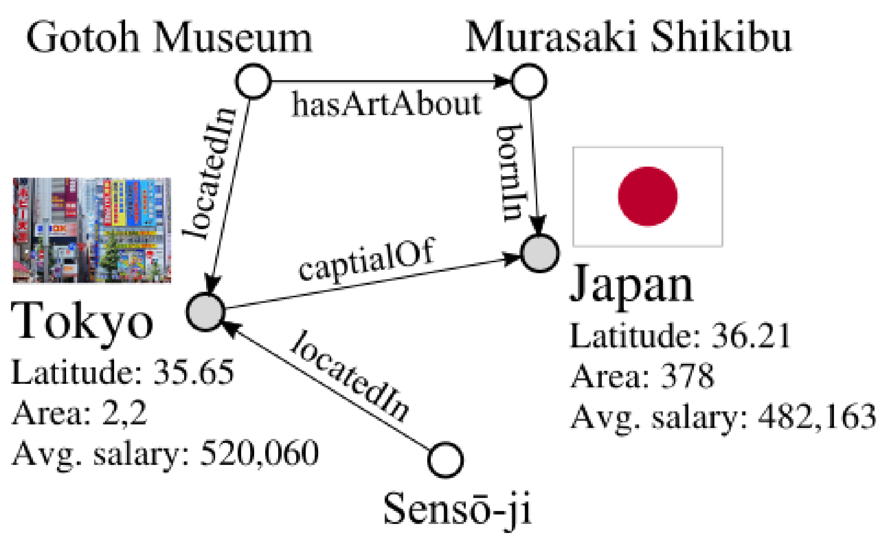
	FB15k-237			
	MR	MRR	Hits@1	Hits@10
TRANSE				
DISTMULT[27]	-	19.1	10.6	37.6
COMPLEX[27]	-	20.1	11.2	38.8
NODE+LINKFEAT[29]	-	23.7	-	36.0
R-GCN+[27]	-	24.8	15.3	41.7
CONVE[4]	330	30.1	22.0	45.8
KBL	231	30.1	21.4	47.5
KBR	2518	18.0	12.8	28.5
KBLR	231	30.6	**22.0**	48.2

Simple Link Prediction in KGs

- Numerical information is very common in knowledge bases (DBpedia, Freebase, YAGO, etc.)
- Examples: geocoordinates, elevation, area, birth year, …

Tokyo

Area	
• **Metropolis**	2,187.66 km^2 (844.66 sq mi)
• **Metro**	13,572 km^2 (5,240 sq mi)
Area rank	45th
Elevation	40 m (130 ft)
Population (July 31, 2016)[3]	
• **Metropolis**	13,617,445
• **Density**	6,224.66/km^2 (16,121.8/sq mi)
• **Metro**	37,800,000
• **Metro density**	2,662/km^2 (6,890/sq mi)
• **23 Wards**	8,967,665 (2015 per prefectural government)
Demonym(s)	江戸っ子 (Edokko), 東京人 (Tokyo-jin), 東京っ子 (Tokyokko), Tokyoite
GDP (Nominal; 2014)[4][5]	
• **Total**	$2 trillion
• **Per capita**	$70,000

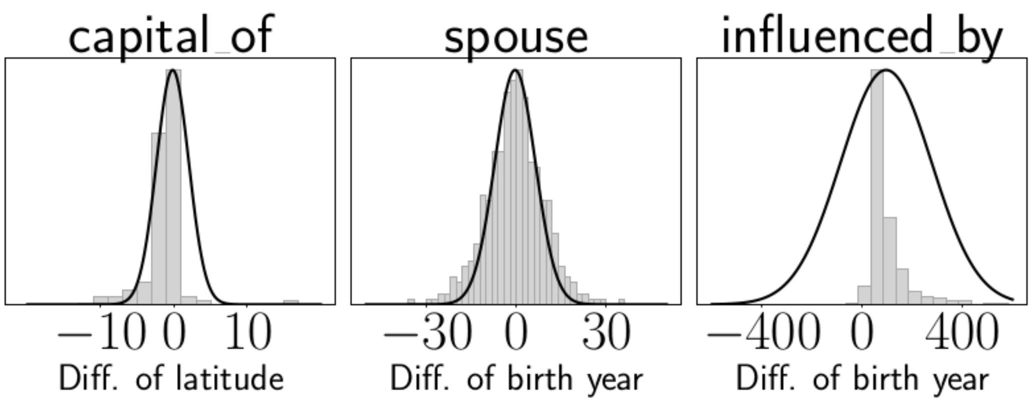


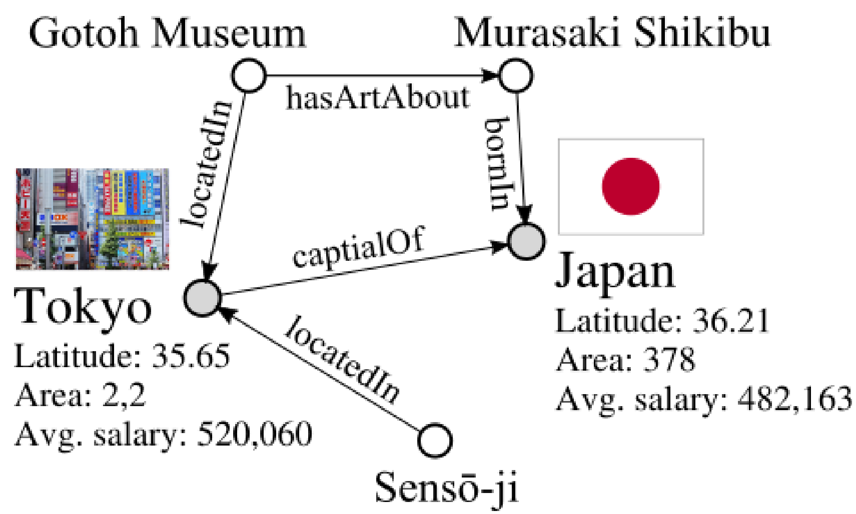
Simple Link Prediction in KGs

How to learn from them?

- Concatenate everything into a vector and pass it to whatever NN ☹

Observation: even though numerical features are not distributed according to a normal distribution, usually the difference between the *head* and *tail* arguments is





- We use the difference values $n_{(h,t)}$ and the fact that they often follow a normal distribution. Why?

$n_{(h,t)} = n_h - n_t$ $\quad\quad$ $N(c, \sigma) = n_h - n_t$ $\quad\quad$ $n_t = n_h + N(c, \sigma)$

- Numerical Expert:

$$f_{(\mathtt{r},\mathtt{N})}(\mathbf{d} \mid \theta_{(\mathtt{r},\mathtt{N})}) = \exp\left(\phi\left(\mathbf{n}_{(\mathtt{h},\mathtt{t})}\right) \cdot \mathbf{w}^{\mathtt{r}}_{\mathtt{num}}\right)$$

where

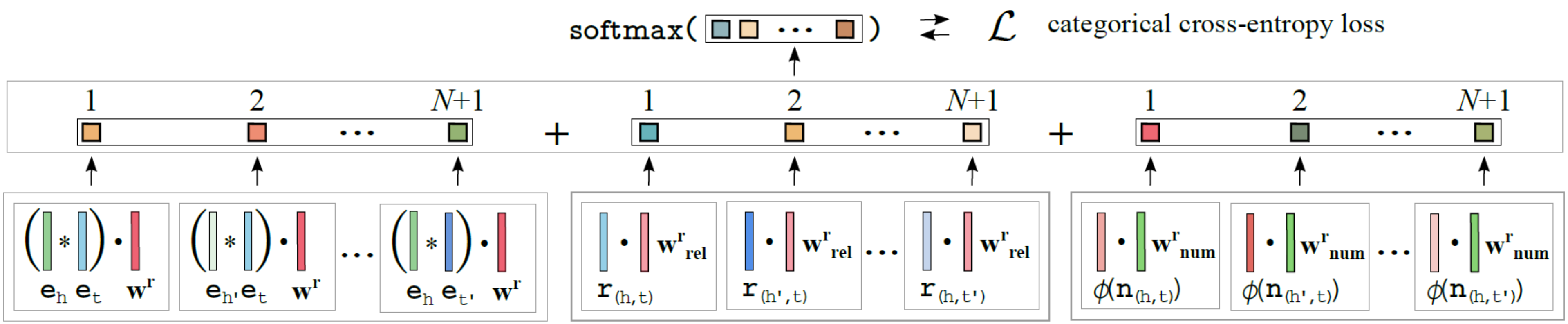
$$\phi\left(\mathbf{n}^{(i)}_{(\mathtt{h},\mathtt{t})}\right) = \exp\left(\frac{-||\mathbf{n}^{(i)}_{(\mathtt{h},\mathtt{t})} - c_i||^2_2}{\sigma^2_i}\right)$$

- Learning from the residual of the underlying linear regression model!

- The output of the RBF is a value between 0 and 1

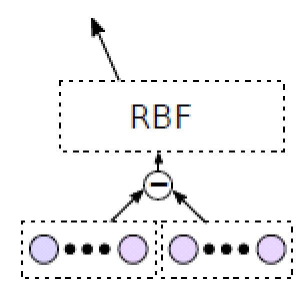
KBlrn: Training

- Simple extension of previous model
- Difference of numerical features + RBF layer
- All parameters of the model are learned end-to-end



$$p(\mathtt{d} \mid \theta_1, \ldots, \theta_n) = \frac{\prod_{\mathtt{F} \in \{\mathtt{R},\mathtt{L},\mathtt{N}\}} f_{(\mathtt{r},\mathtt{F})}(\mathtt{d} \mid \theta_{(\mathtt{r},\mathtt{F})})}{\sum_{\mathtt{c}} \prod_{\mathtt{F} \in \{\mathtt{R},\mathtt{L},\mathtt{N}\}} f_{(\mathtt{r},\mathtt{F})}(\mathtt{c} \mid \theta_{(\mathtt{r},\mathtt{F})}))}$$

$$\exp\left(\frac{-\|\mathbf{n}^{(i)}_{(\mathtt{h},\mathtt{t})} - c_i\|_2^2}{\sigma_i^2}\right)$$



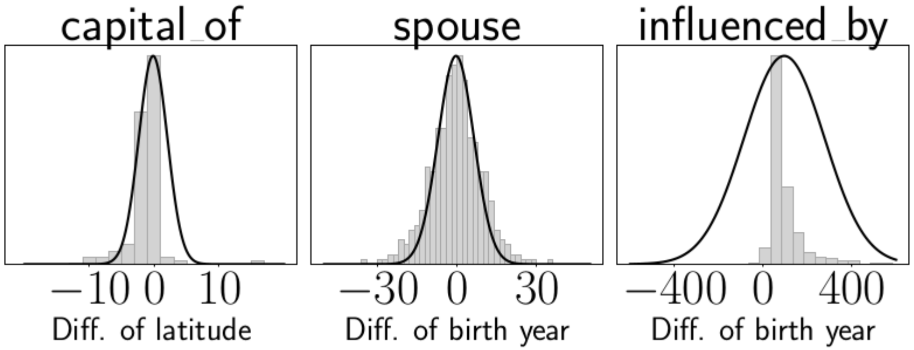
Simple Link Prediction in KGs
KBlrn: Performance

Numerical features helpful for KB completion

Metrics:
MR: Mean rank of correct triple
MRR: Mean reciprocal rank
Hits@1: Percentage of correct triples ranked 1
Hits@10: Percentage of correct triples ranked in the tope 10

	FB15k-num				FB15k-237-num			
	MR	MRR	Hits@1	Hits@10	MR	MRR	Hits@1	Hits@10
TRANSE								
DISTMULT	39	72.6	62.1	89.7	195	26.4	16.4	47.3
	without numerical features							
KBL	39	72.6	62.1	89.7	195	26.4	16.4	47.3
KBR	399	84.7	**81.6**	90.1	3595	23.6	17.8	36.1
KBLR	28	85.3	80.3	92.4	232	29.3	19.7	49.2
	with numerical features							
KBLN	32	73.6	63.0	90.7	122	28.6	17.9	51.6
KBRN	68	84.0	80.6	90.0	600	26.1	19.3	39.7
KBLRN	**25**	**85.9**	81.0	**92.9**	**121**	**31.4**	**21.2**	**52.3**

- Let's have a deeper look at the relational expert:



	KBLR		KBLRN	
Relation	MRR	H@10	MRR	H@10
capital_of	5.7	13.6	**14.6**	**18.2**
spouse_of	4.4	**0.0**	**7.9**	**0.0**
influenced_by	7.3	20.9	**9.9**	**26.8**

- What happens if we use a function other than a RBF:

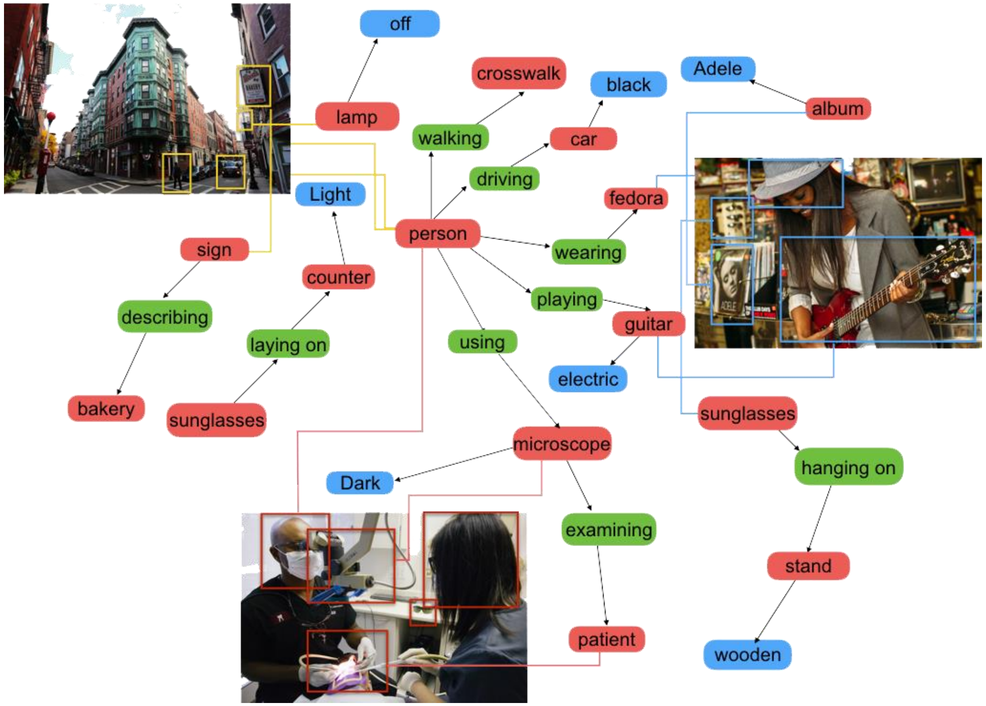
ϕ	MRR	Hits@10
sign	29.7	50.1
RBF	**31.4**	**52.3**

Outline

- Quick Reminder
- Failing with Latent and Relational Models
- **Simple Link Prediction in KGs:**
  - Graph Structure and Numerical Information
  - **Visual Information**
- Simple Link Prediction in Temporal KGs
- Remarks

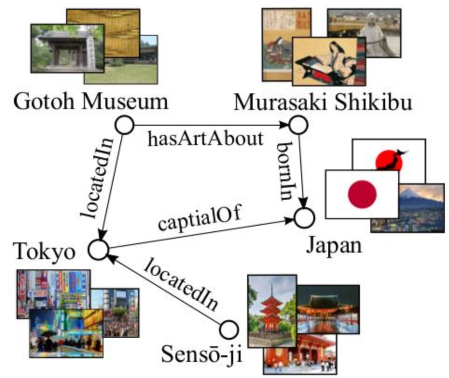
- What about a KG of visual data?

- VisualGenome [Krishna et al.,2015]



Simple Link Prediction in KGs

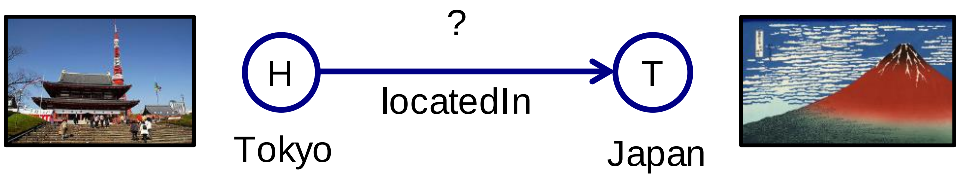
- A KG where relationships hold between pairs of images?



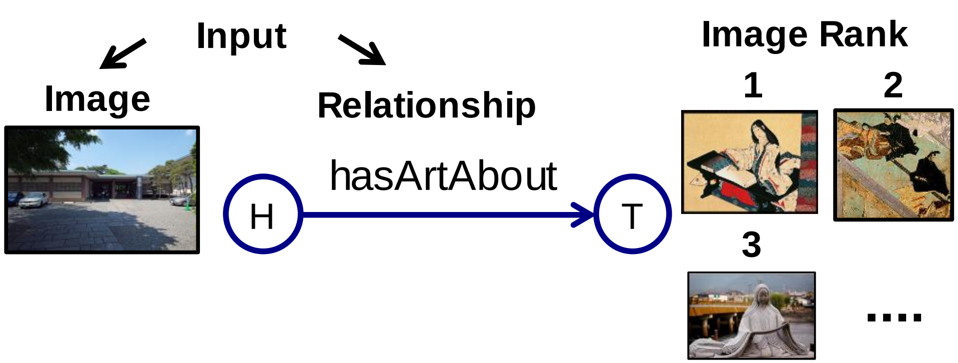
- We crawled images from search Engines using FB as a blueprint

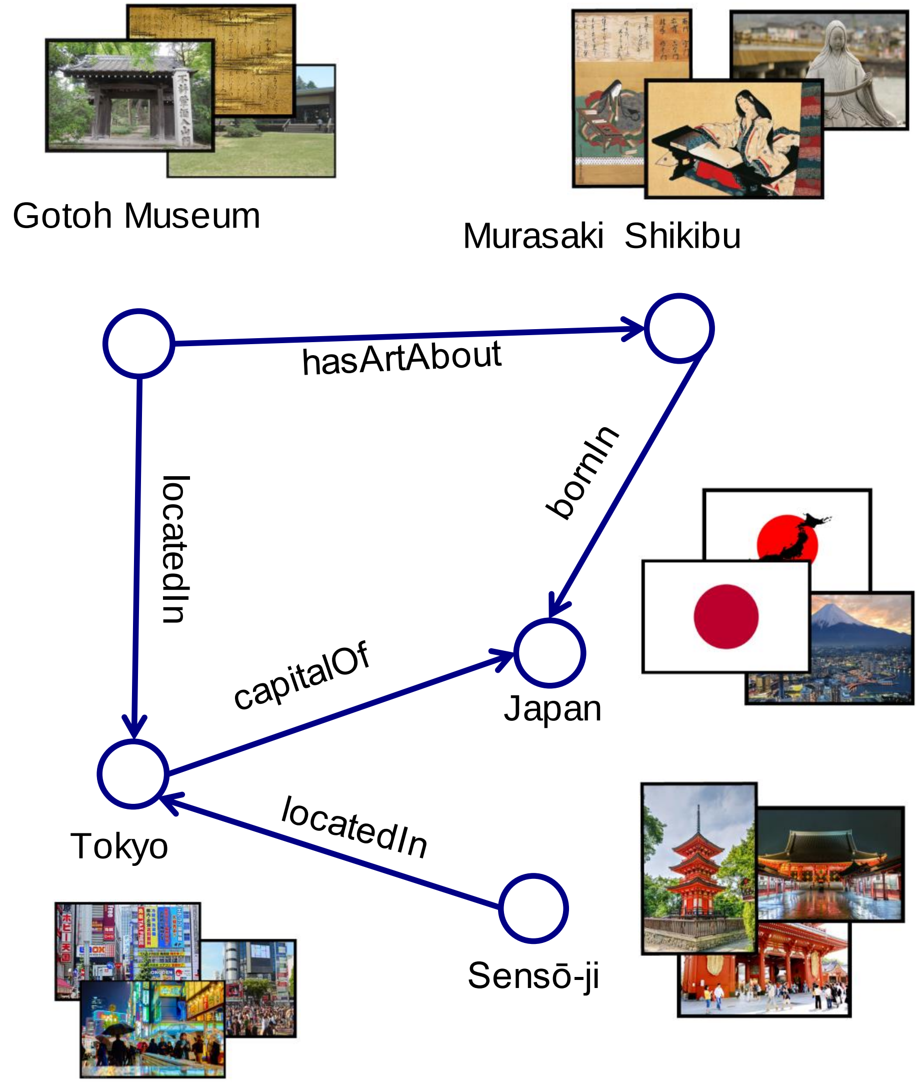
DB	Entities	Relations	Triplets			Images		
			Train	Validation	Test	Train	Validation	Test
ImageNet	21.841	18		-			14.197.122	
VisualGenome	75.729	40.408		1.531.448			108.077	
FB15k	14.951	1.345	483.142	50.000	59.071	0	0	0
ImageGraph	14.870	1.330	460.406	47.533	56.071	411.306	201.832	216.793

1 Given two unseen images, predict the relationship



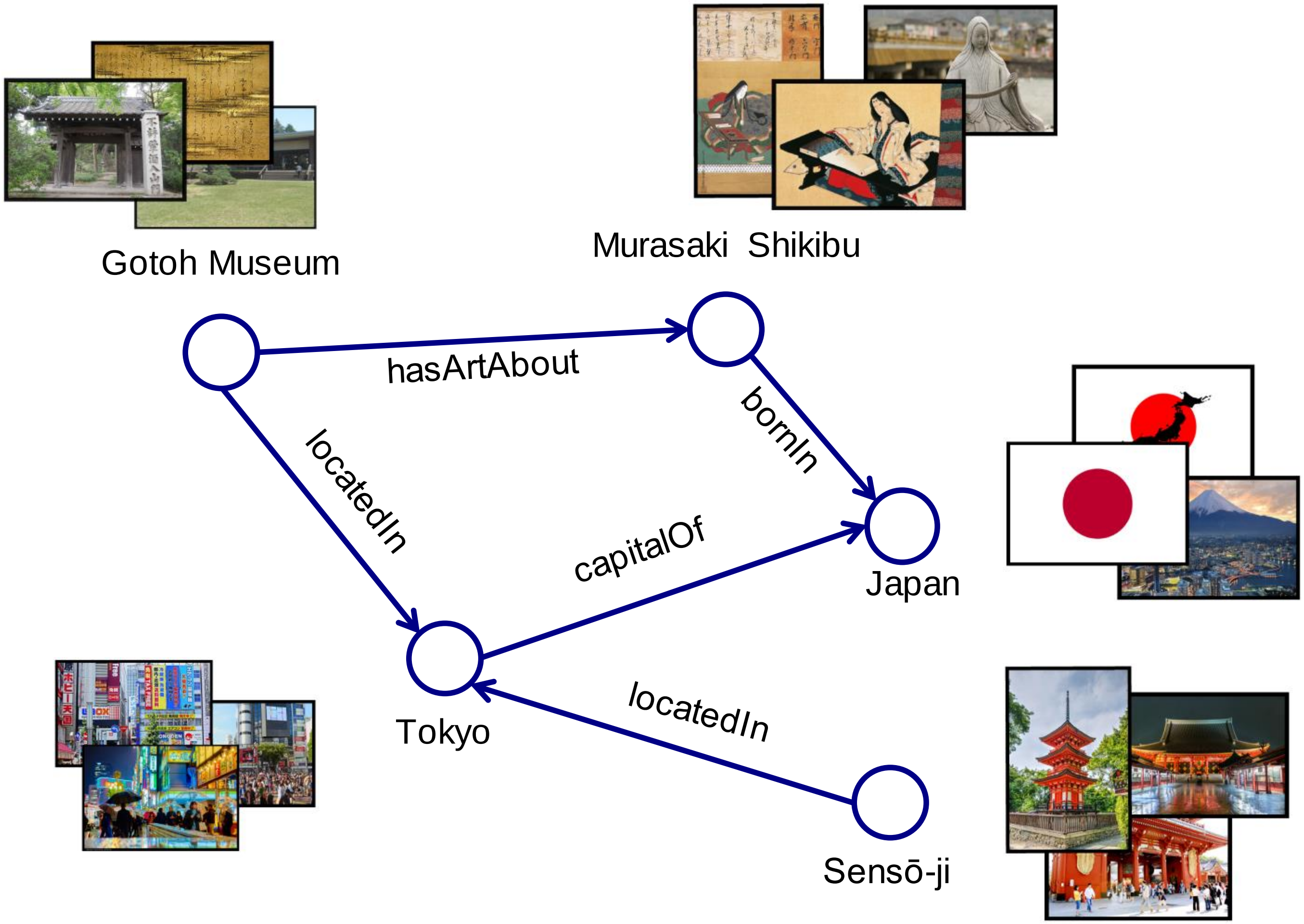
2 Given an unseen image and a relationship, retrieve related images



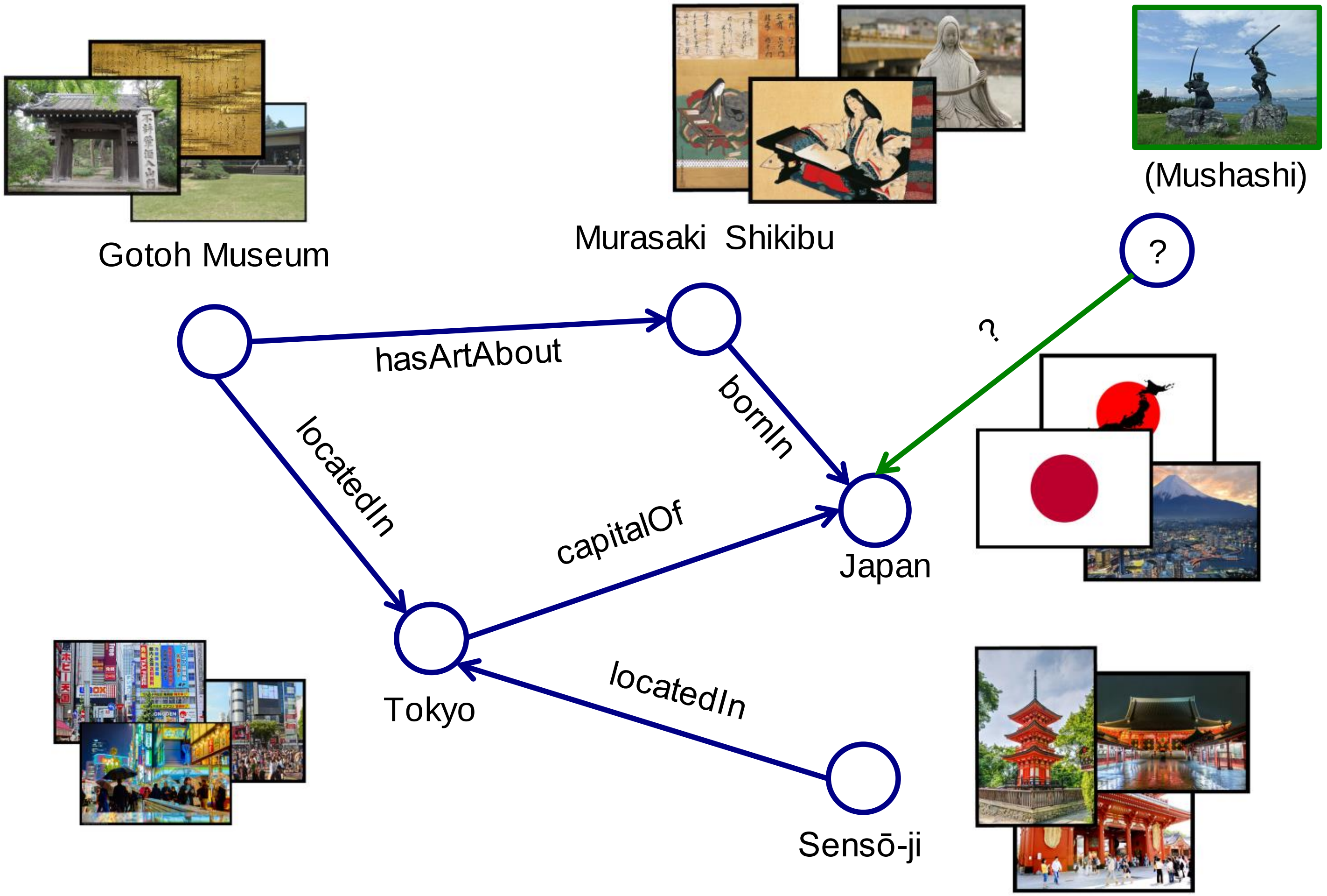


Simple Link Prediction in KGs
Query Types

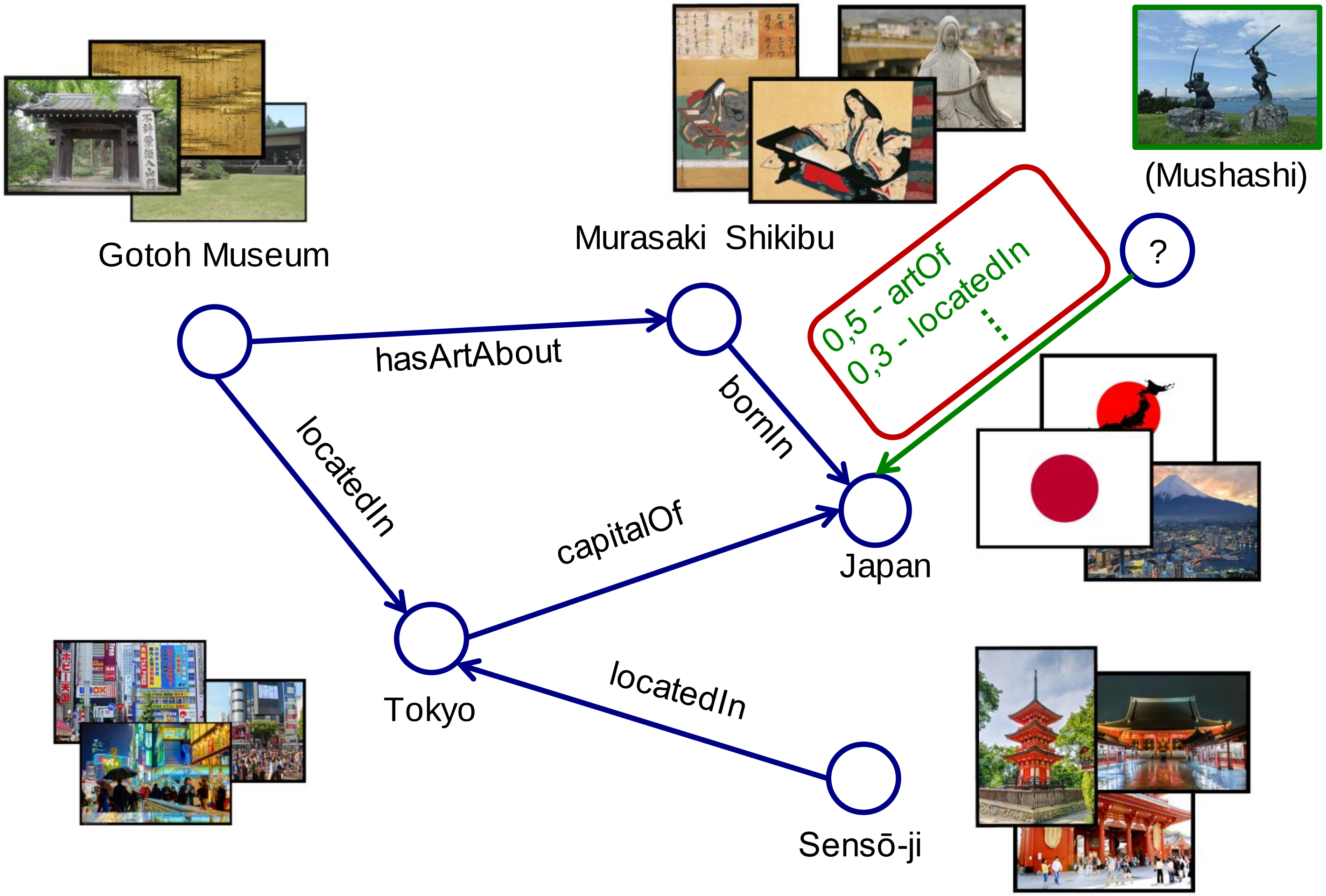
Zero-Shot: predicting relationships for new entities



Zero-Shot: predicting relationships for new entities

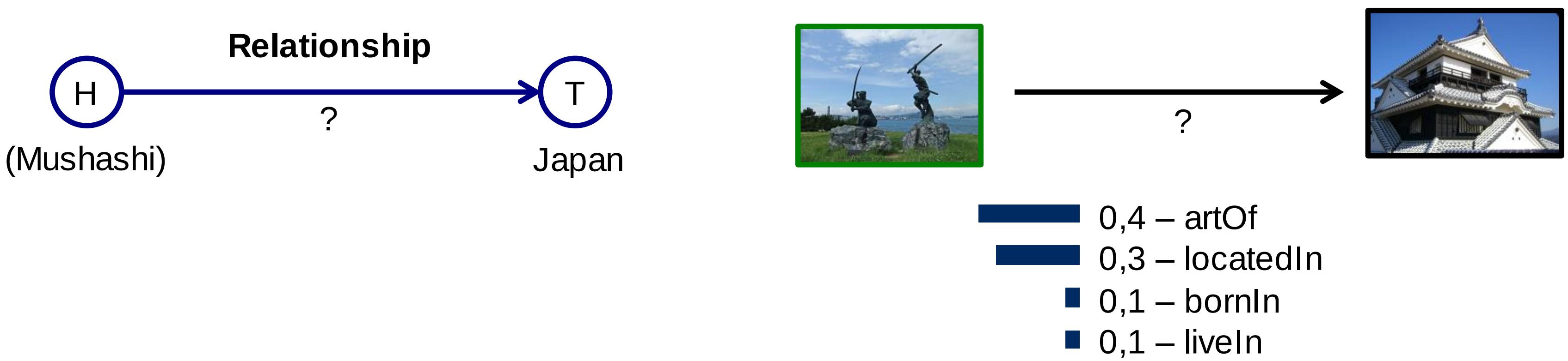


Zero-Shot: predicting relationships for new entities



Zero-Shot: predicting relationships for new entities

3 Unseen image -> Unseen image

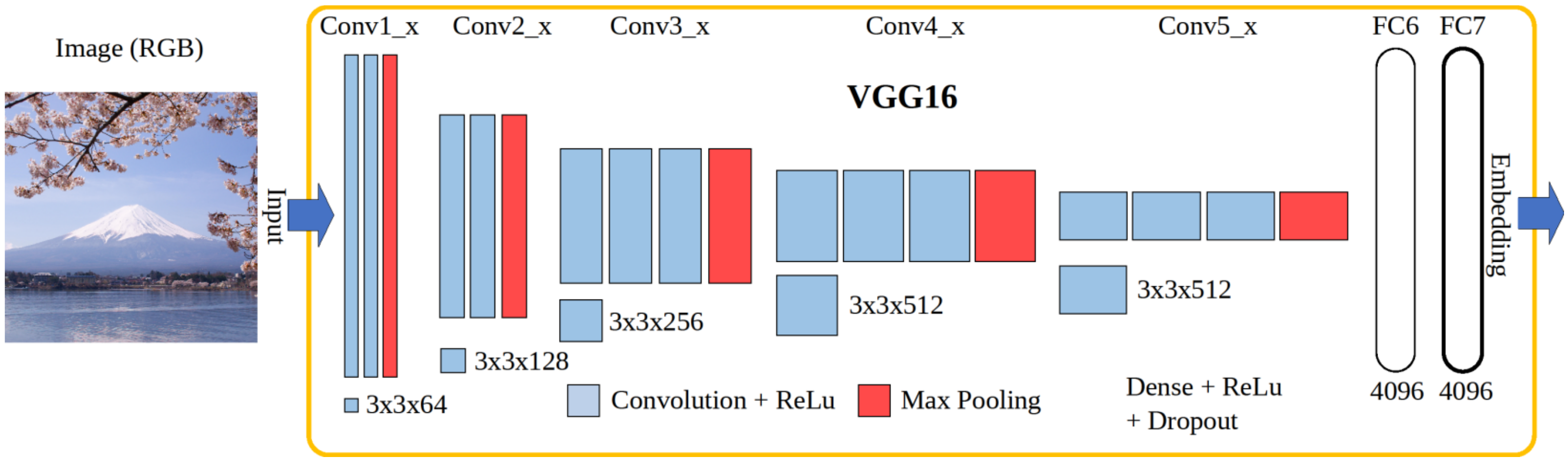


4 Unseen image -> Seen images

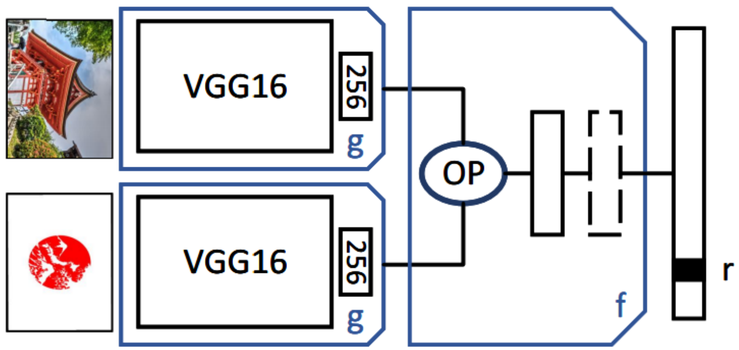
Simple Link Prediction in KGs

Image Graph [Onoro-Rubio et al.,2017]

- Acknowledgement: [Lonij et al.,2017]
- Function that learns latent representations from images



- Standard Scoring Functions



Simple Link Prediction in KGs

ImageGraph: Performance

- Given a pair of **unseen images** for which we do not know their KG entities, determine the **relations** between these underlying entities

1

?

Relationship

Performance Results:

Model	Median	Hits@1	Hits@10	MRR
Baseline	35	3.7	26.5	0.104
DIFF	11	21.1	50.0	0.307
MULT	8	15.5	54.3	0.282
CAT	**6**	**26.7**	**61.0**	**0.378**
DIFF+1HL	8	22.6	55.7	0.333
MULT+1HL	9	14.8	53.4	0.273
CAT+1HL	**6**	25.3	60.0	0.365

Probability based

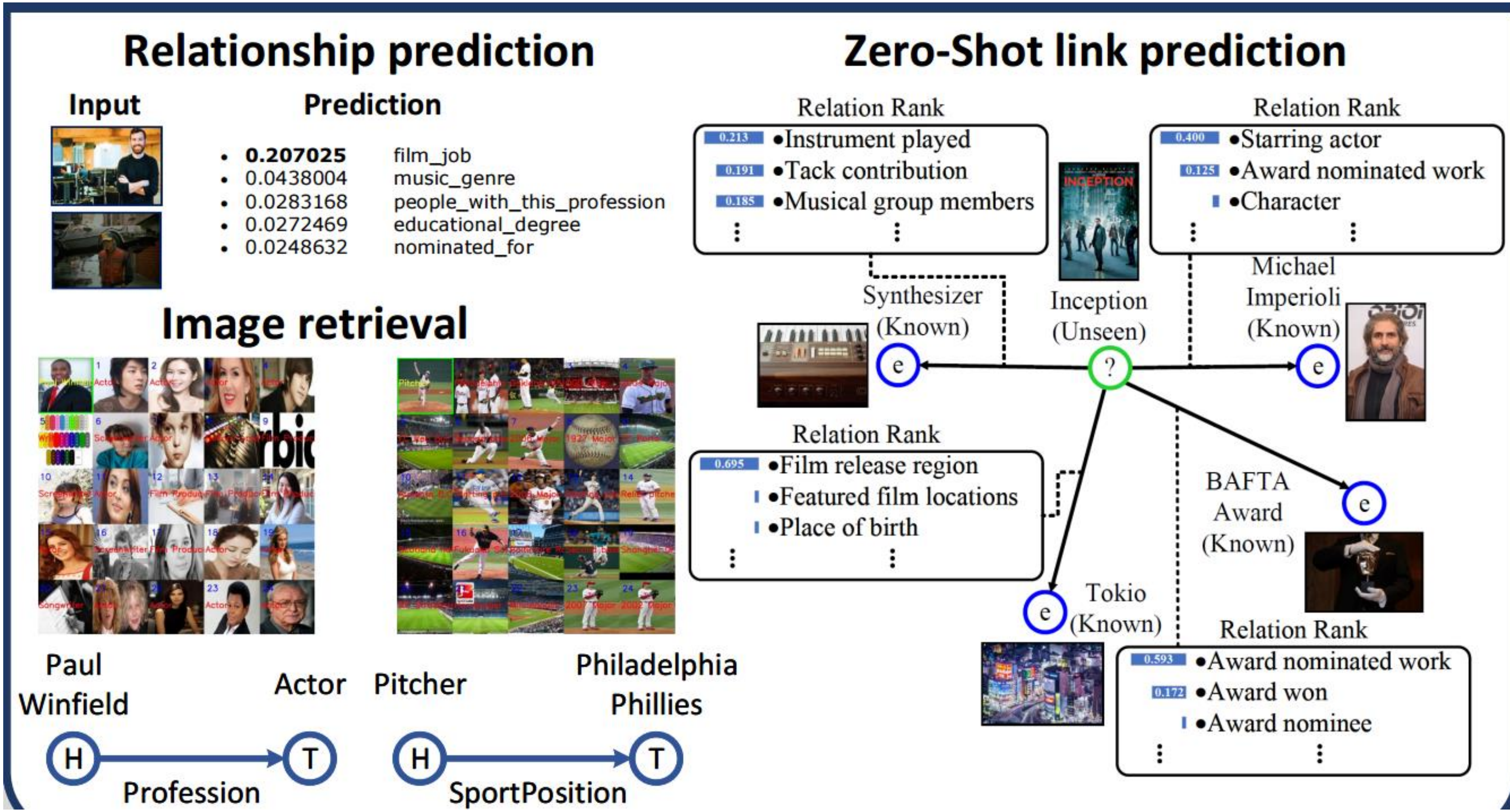
Metrics:
Median – Median rank of the correct answer
Hits@1 – Percentage of correct answers ranked top 1
Hits@10 – Percentage of correct answers ranked in the tope 10
MRR – Mean reciprocal rank

+ 1HL (additional hidden layer)

Composition function *op*: difference (DIFF), multiplication (MULT), concatenation (CAT)

Simple Link Prediction in KGs

ImageGraph: Qualitative Results



Advantages & Disadvantages of:

- Latent models
- Relational models

KBlrn:

- A learning framework that combines multiple experts
- Reasoning with numerical information

Image Graph:

- A KG enriched with visual information
- Novel visual query types
- Zero-shot learning

5/10 minutes break

Outline

- Quick Reminder
- Failing with Latent and Relational Models
- Simple Link Prediction in KGs:
  - Graph Structure and Numerical Information
  - Visual Information
- **Simple Link Prediction in Temporal KGs**
- Remarks

Simple Link Prediction in Temporal KGs

Research on link prediction has mainly focused on static KGs.

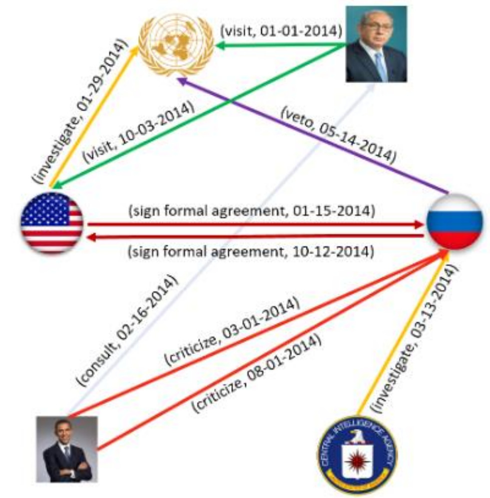
- What about time information?

Freebase, Yago, DBpedia… They all have time information in the original dumps

yagoDateFacts All facts of YAGO that contain dates	Preview	Download TTL	Download TSV
yagoFacts All facts of YAGO that hold between instances	Preview	Download TTL	Download TSV

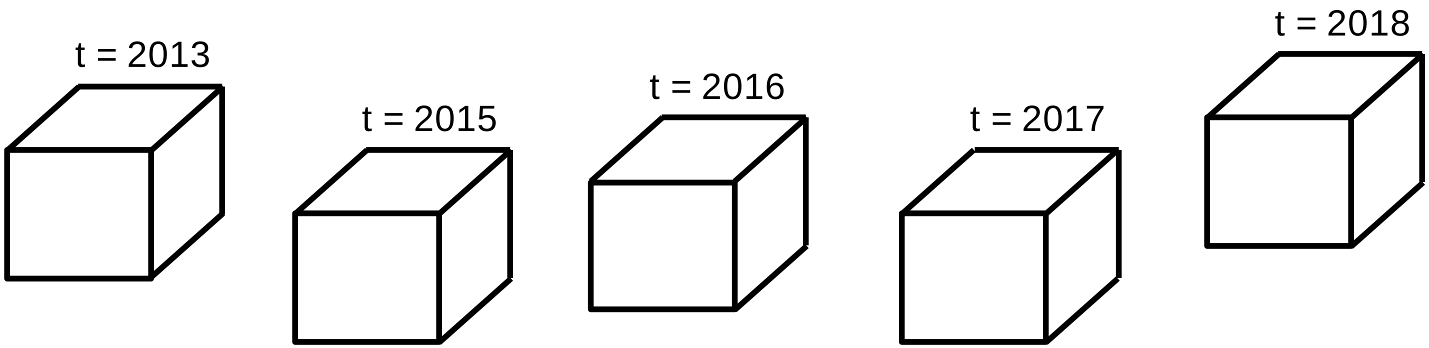
Example of Temporal KG

```
<Jamie_Lawrence>        <playsFor>      <Sunderland_A.F.C.>     <occursUntil>   "1994-##-##"
<Jeph_Loeb>     <created>       <Lost_(TV_series)>
<Bradley_Johnson>       <playsFor>      <Leeds_United_F.C.>     <occursSince>   "2008-##-##"
<Bradley_Johnson>       <playsFor>      <Leeds_United_F.C.>     <occursUntil>   "2011-##-##"
<David_Hibbert> <playsFor>      <Shrewsbury_Town_F.C.>  <occursSince>   "2007-##-##"
<David_Hibbert> <playsFor>      <Shrewsbury_Town_F.C.>  <occursUntil>   "2010-##-##"
<George_Eastham,_Sr.>   <isAffiliatedTo>        <York_City_F.C.>
<Marcus_Gayle>  <playsFor>      <Rangers_F.C.>  <occursSince>   "2001-##-##"
<Iván_Campo>    <playsFor>      <AEK_Larnaca_FC>
```



Simple Link Prediction in Temporal KGs

Multiple extremely sparse tensors



Not too much work…

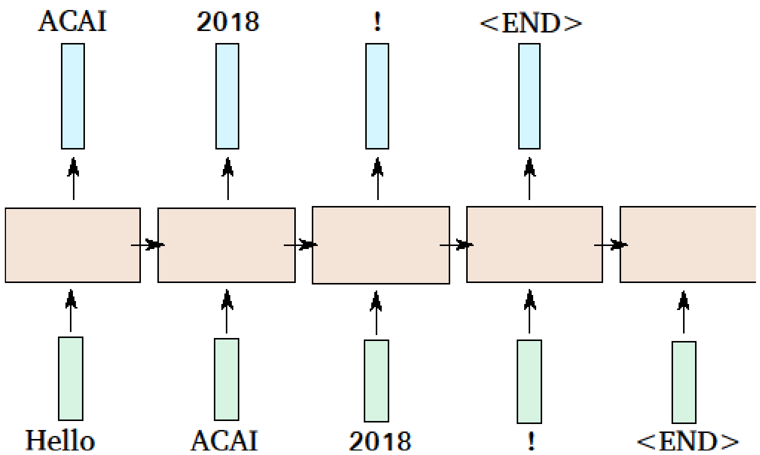
- Temporal logic [van Benthem, 1995]
- Latent models [Jiang et al.,2016; Trivedi et al.,2017;Leblay et al.,2018]

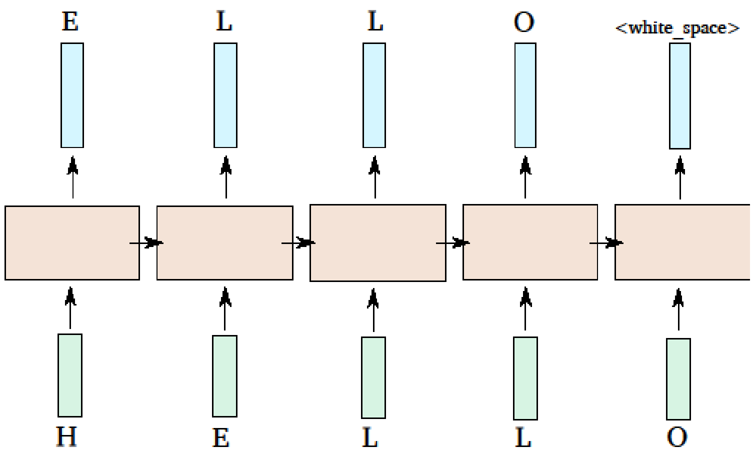
Challenges:

- Time information is very sparse
- Heterogeneity of time information
- Point-in-Time / Intervals of time

Simple Link Prediction in Temporal KGs

(Not really) Off-topic: Language Modeling





RNN Architecture
One representation per word
Limited vocabulary

Char - RNN Architecture
One representation per character
Unlimited vocabulary

Simple Link Prediction in Temporal KGs

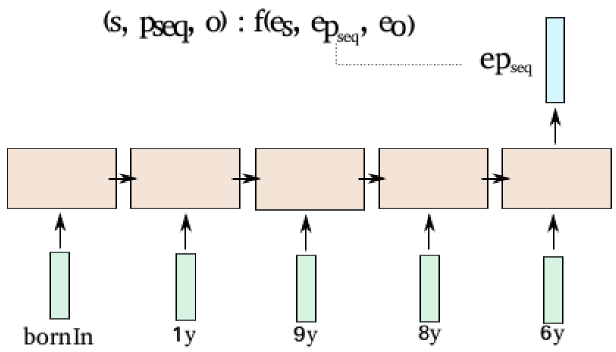
- [Garcia-Duran et al.,2018] Incorporate time information into standard embedding approaches for link prediction like

$$\text{TRANSE: } f(s,p,o) = ||\mathbf{e}_s + \mathbf{e}_p - \mathbf{e}_o||_2$$
$$\text{DISTMULT: } f(s,p,o) = (\mathbf{e}_s \circ \mathbf{e}_o)\mathbf{e}_p^T.$$

- Trick:

Fact	Predicate Sequence
(BO, country, US)	[country]
(BO, bornIn, US, 1961)	[born, $1y$, $9y$, $6y$, $1y$]
(BO, president, US, since, 2009-01)	[president, since, $2y$, $0y$, $0y$, $9y$, $01m$]

- Architecture:



$$\text{TA-TRANSE: } f(s,p_{seq},o) = ||\mathbf{e}_s + \mathbf{e}_{p_{seq}} - \mathbf{e}_o||_2$$
$$\text{TA-DISTMULT: } f(s,p_{seq},o) = (\mathbf{e}_s \circ \mathbf{e}_o)\mathbf{e}_{p_{seq}}^T.$$

Simple Link Prediction in Temporal KGs

Temporal KGs

Data set	YAGO15K	ICEWS '14	ICEWS 05-15	WIKIDATA
Entities	15,403	6,869	10,094	11,134
Relationships	34	230	251	95
#Facts	138,056	96,730	461,329	150,079
#Distinct TS	198	365	4,017	328
Time Span	1513-2017	2014	2005-2015	25-2020
Training	110,441 [29,381]	78,826 [78,826]	368,962 [368,962]	121,422 [121,422]
Validation	13,815 [3,635]	8,941 [8,941]	46,275 [46,275]	14,374 [14,374]
Test	13,800 [3,685]	8,963 [8,963]	46,092 [46,092]	14,283 [14,283]

Queries

- (s, p, ?) / (s, p, ?, date) / (s, p, ?, temporal_modifier, date)
- (?, p, o) / (?, p, o, date) / (?, p, o, temporal_modifier, date)

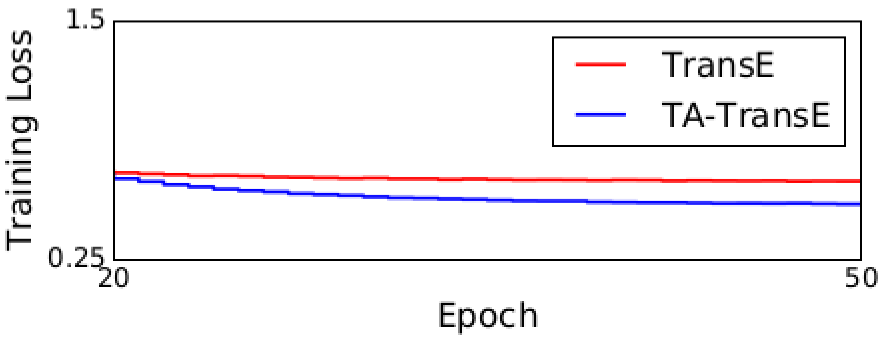
Standard Evaluation Metrics

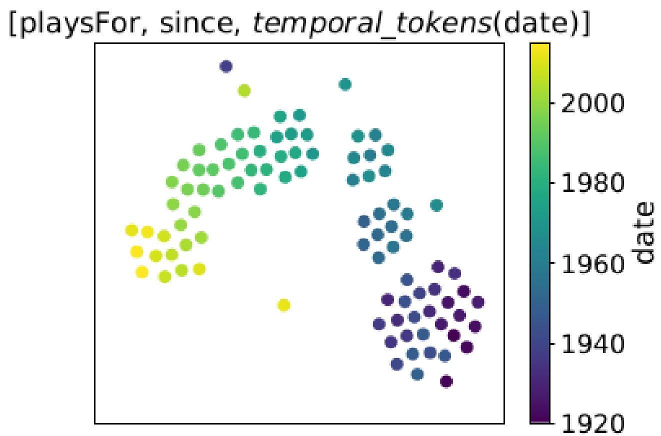
- MRR, MR, Hits@1, Hits@10

Simple Link Prediction in Temporal KGs

	YAGO15K				WIKIDATA			
	MRR	MR	Hits@10	Hits@1	MRR	MR	Hits@10	Hits@1
TTRANSE	**32.1**	578	51.0	23.0	48.8	80	80.6	33.9
TRANSE	29.6	614	46.8	22.8	31.6	**50**	65.9	18.1
DISTMULT	27.5	578	43.8	21.5	31.6	77	66.1	18.1
TA-TRANSE	**32.1**	564	**51.2**	**23.1**	48.4	79	**80.7**	32.9
TA-DISTMULT	29.1	**551**	47.6	21.6	**70.0**	198	78.5	**65.2**

	ICEWS 2014				ICEWS 2005-15			
	MRR	MR	Hits@10	Hits@1	MRR	MR	Hits@10	Hits@1
TTRANSE	25.5	148	60.1	7.4	27.1	181	61.6	8.4
TRANSE	28.0	**122**	63.7	9.4	29.4	84	66.3	9.0
DISTMULT	43.9	189	67.2	32.3	45.6	90	69.1	33.7
TA-TRANSE	27.5	128	62.5	9.5	29.9	**79**	66.8	9.6
TA-DISTMULT	**47.7**	276	**68.6**	**36.3**	**47.4**	98	**72.8**	**34.6**





t-SNE representations of a predicate sequence for different dates

- All data sets are available at:

https://github.com/nle-ml/mmkb

Outline

- Quick Reminder
- Failing with Latent and Relational Models
- Simple Link Prediction in KGs:
  - Graph Structure and Numerical Information
  - Visual Information
- Simple Link Prediction in Temporal KGs
- **Remarks**

A very important evaluation problem

- Remember KGs are far from complete…
- … but they are used as ground truth for completeness evaluation
- FB15k: (US, contains, ?)
  - Around 1600 entities answer that query
  - Less than 1000 of these correct completions are in the ground truth

Advice: Run exps. in many data sets

Importance of the loss & hyperparameter choices [Kadlec et al.,2017]

DistMult (orig) (Yang et al., 2015)	-	94.2	0.83	-	57.7	0.35
DistMult (Toutanova and Chen, 2015)	-	-	-	-	79.7	0.555
DistMult (Trouillon et al., 2017)	-	93.6	0.822	-	82.4	0.654
Single DistMult (this work)	655	94.6	0.797	42.2	**89.3**	**0.798**

You can learn a lot!

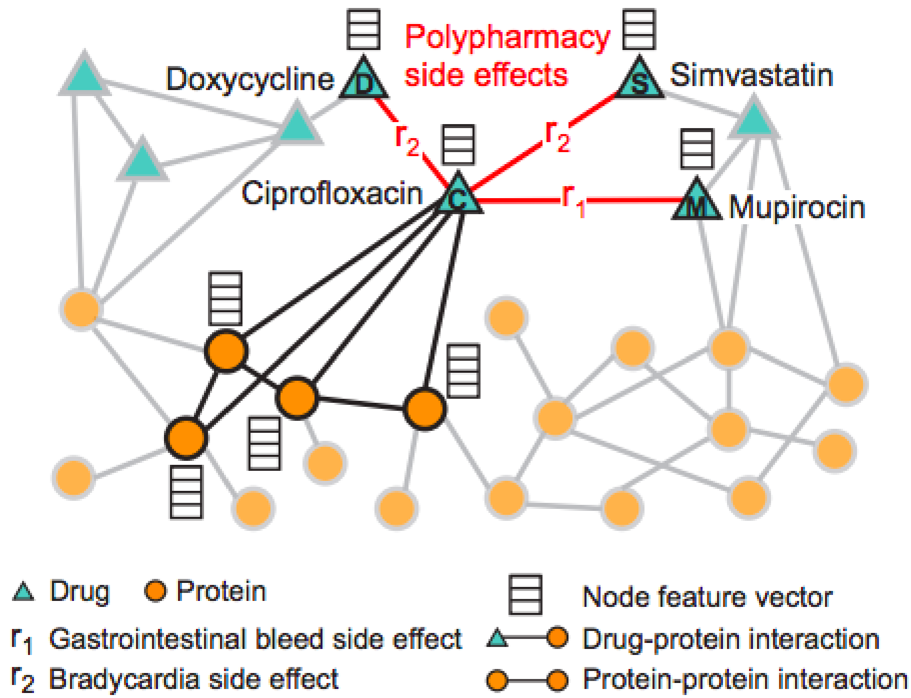
- Experiments are relatively fast
- Implementation of these methods is easy

```
# distMult model
e1 = Input(shape=(num_negative + 1,), name="e1")
e2 = Input(shape=(1,), name="e2")
rel = Input(shape=(1,), name="rel")

# Embeddings
ent_embedding = Embedding(numEnt, embedding_dim, name='ent_embeddings')
rel_embedding = Embedding(numRel, embedding_dim, name='rel_embeddings')

e1_emb = Dropout(drop)(ent_embedding(e1))
e2_emb = Dropout(drop)(ent_embedding(e2))
rel_emb = Dropout(drop)(rel_embedding(rel))

mult_elmWise = Multiply(name='joint_emb')([rel_emb, e2_emb])
score = Dot(2, name='score')([mult_elmWise, e1_emb])

score = Activation('softmax')(Reshape((num_negative + 1,), name='soft_out')(score))
adam = Adam()

model = Model(input=[e1, e2, rel], output=[score])
model.compile(loss='categorical_crossentropy', optimizer=adam)
```

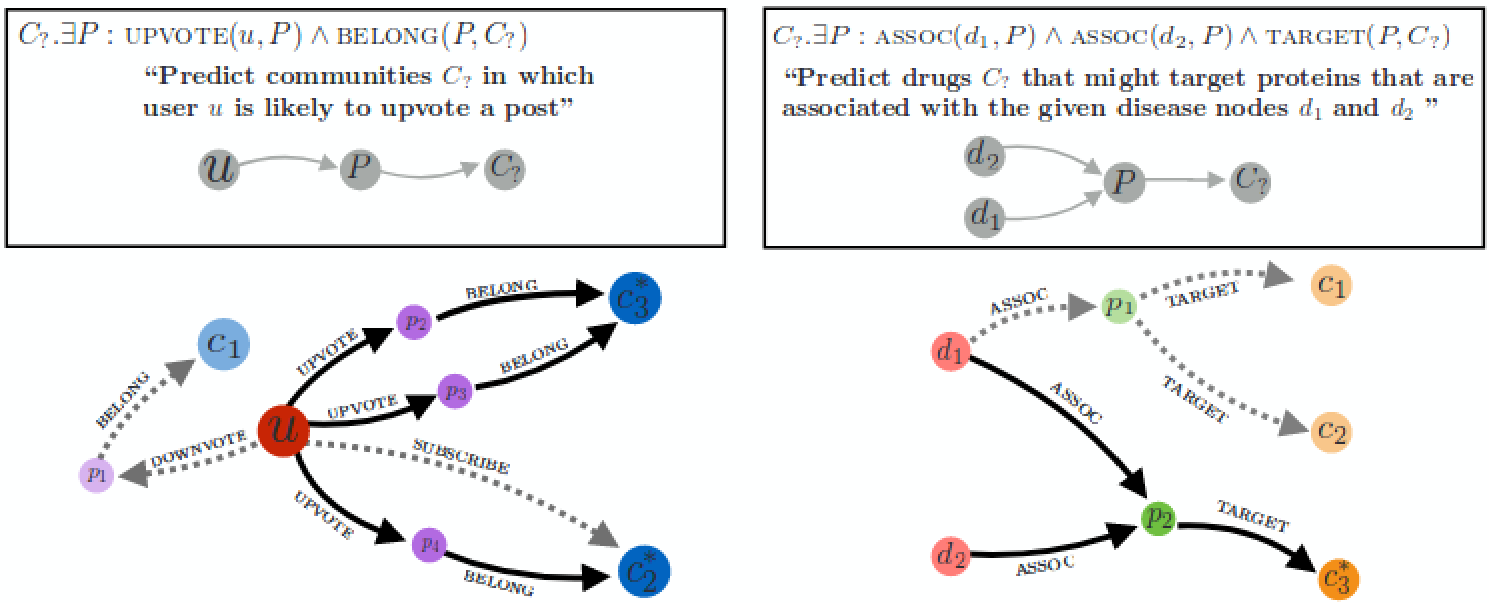
Simple Link Prediction

- Don’t do it
- There are many other interesting related-problems!

Link Prediction in Attributed KGs [Zitnik et al.,2018]



Complex Queries [Hamilton et al.,2018]



Non-Binary Predicates

- ((Alberto, Mathias), met_at, Heidelberg)
- Reification: turn n-ary predicates to binary ones

Entity Linking

- Specific type of query (h, sameAs, ?) / (?, sameAs, t)

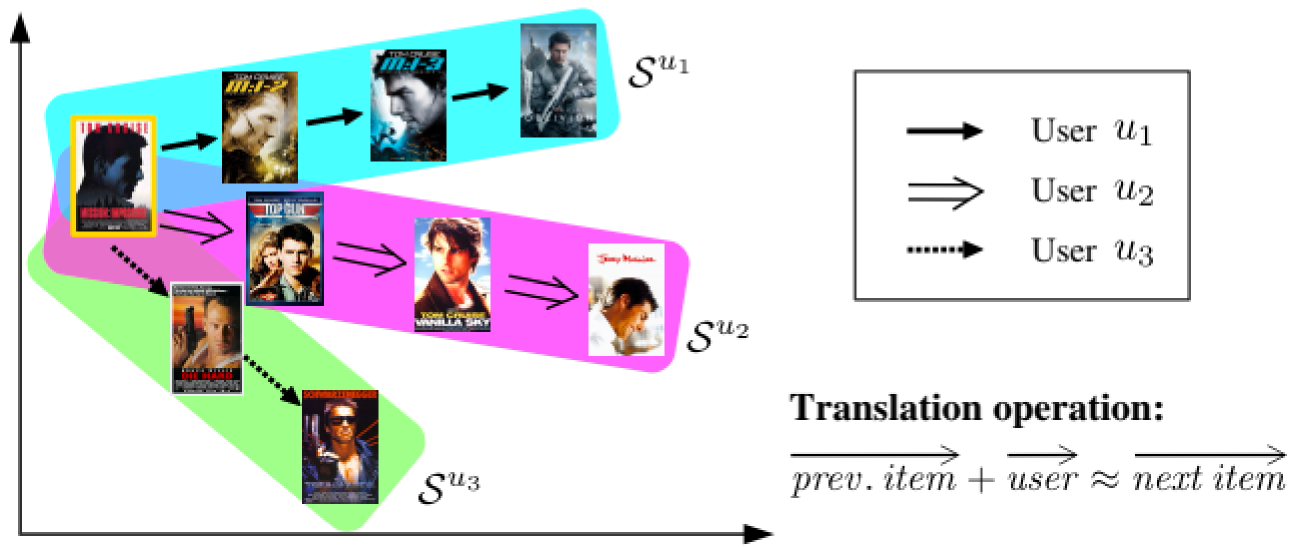
Hierarchical Embeddings in KGs?

- [Nickel et al.,2017] learn hierarchical embeddings for taxonomies

Get inspiration for other problems!

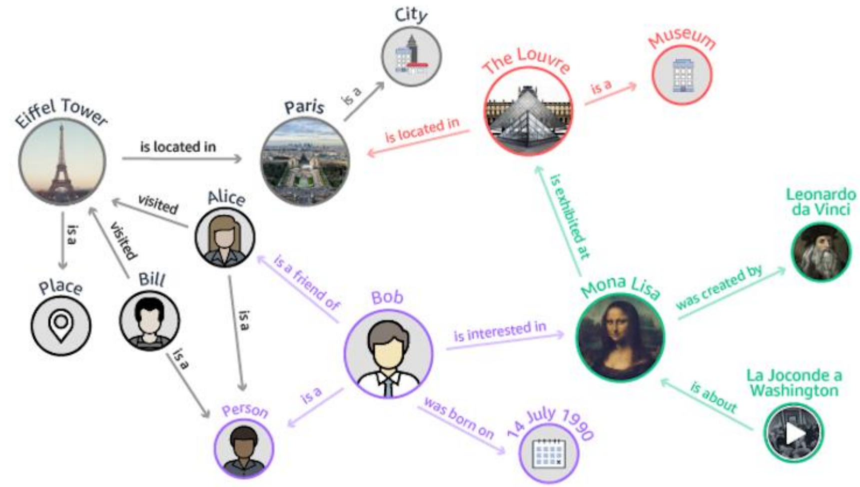
Recommender Systems

- Sequential Recommendation [He et al.,2017]

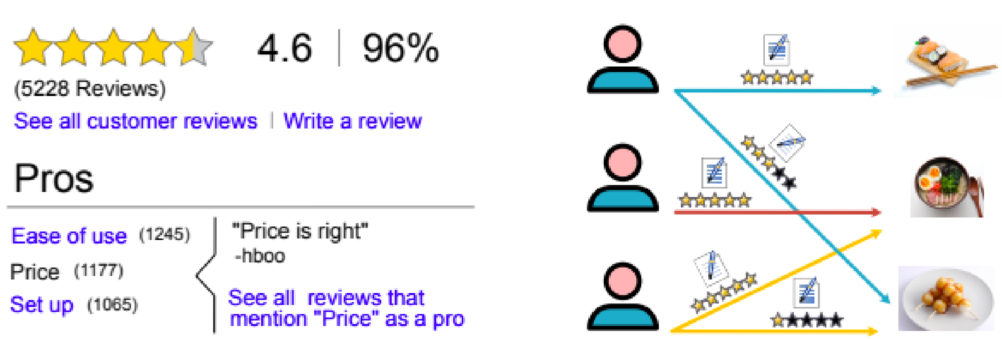


Recommender Systems

- Reviews available [Garcia-Duran et al.,2017b]

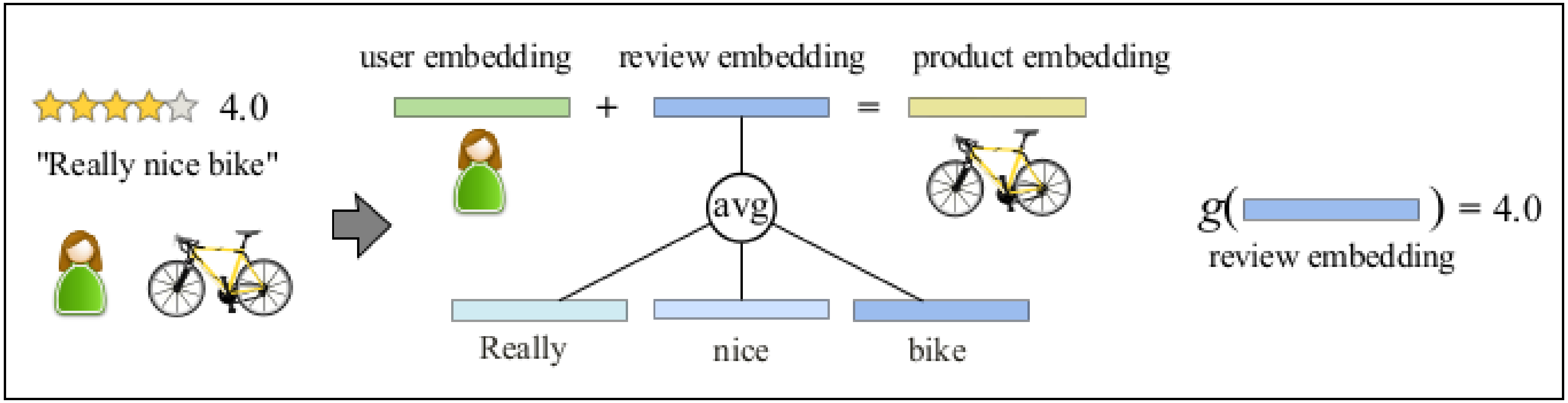


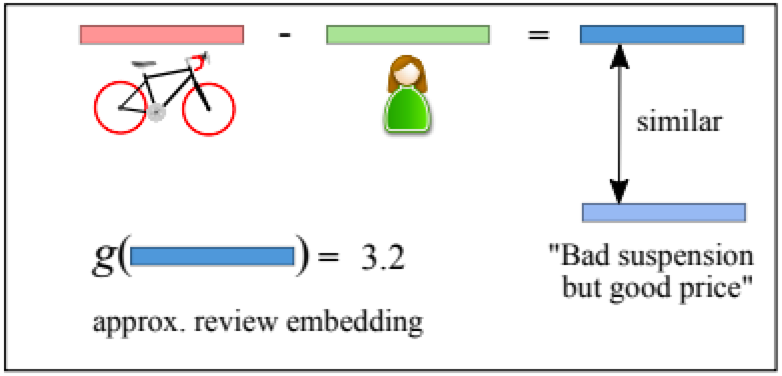
Knowledge Graph



Recommender System

Recommender Systems





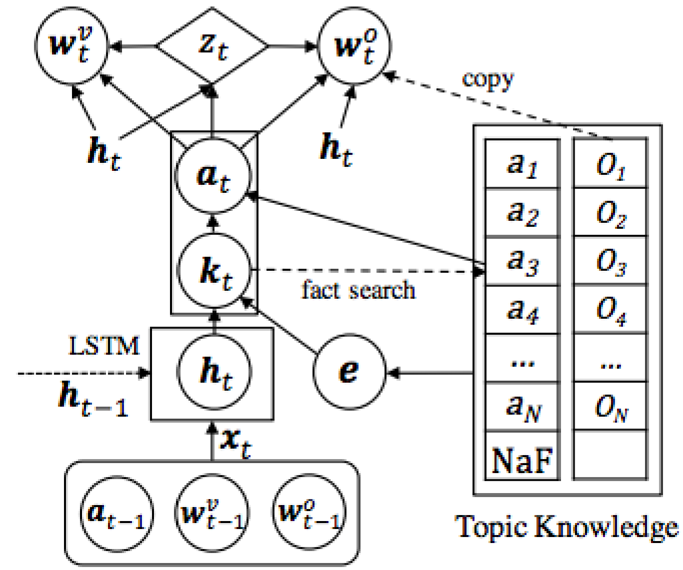
Language Modeling [Ahn et al., 2016]

Example. Given a topic on Fred Rogers with topic description

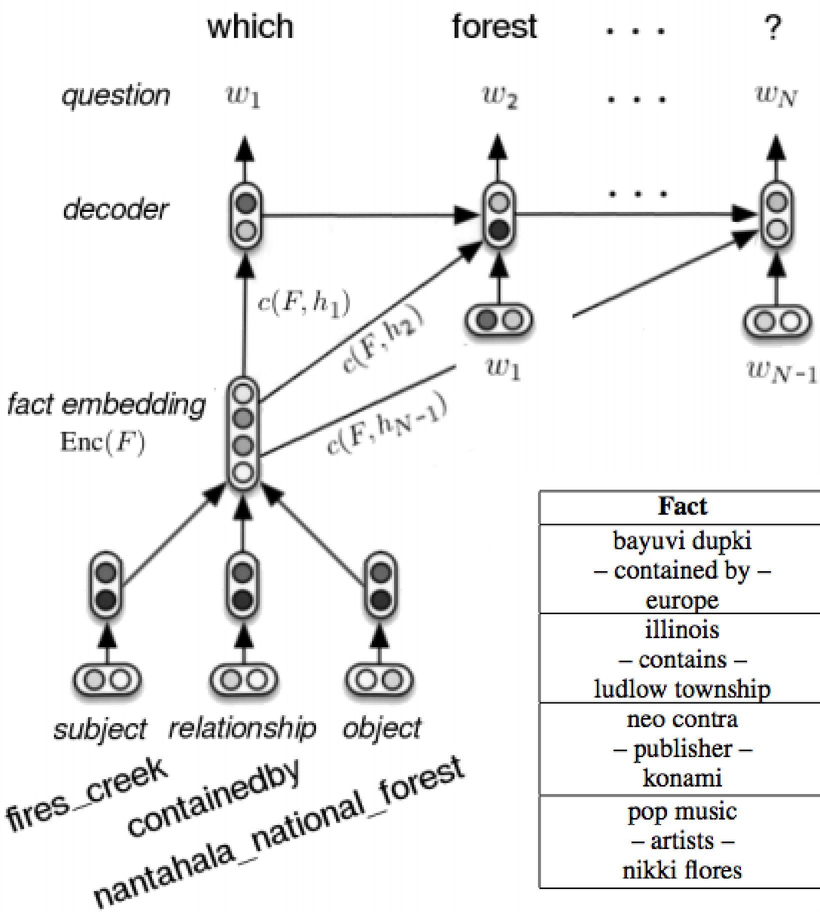
W = "Rogers was born in Latrobe, Pennsylvania in 1928"

and topic knowledge $\mathcal{F} = \{a^{42}, a^{83}, a^{0}\}$ where

a^{42} = *(Fred_Rogers, Place_of_Birth, Latrobe_Pennsylvania)*
a^{83} = *(Fred_Rogers, Year_of_Birth, 1928)*
a^{0} = *(Fred_Rogers, Topic_Itself, Fred_Rogers)*,



Text Generation [Serban et al., 2016]



Fact	Human	Baseline	MP Triples TransE++
bayuvi dupki – contained by – europe	where is bayuvi dupki?	what state is the city of bayuvi dupki located in?	what continent is bayuvi dupki in?
illinois – contains – ludlow township	what is in illinois?	what is a tributary found in illinois?	what is the name of a place within illinois?
neo contra – publisher – konami	who published neo contra?	which company published the game neo contra?	who is the publisher for the computer videogame neo contra?
pop music – artists – nikki flores	what artist is known for pop music?	An example of pop music is what artist?	who's an american singer that plays pop music?

- Learned the basic concepts of:
  - Relational Models
  - Latent Models
  - Strengths & Weaknesses

- How to learn from modalities other than the graph structure

- How to learn with temporal information

- Many interesting problems to address
  - Simple Link Prediction is not one of them

- People are using KGs in many other problems

References

[Ahn et al.,2017] A Neural Knowledge Language Model

[An et al.,2018] Accurate Text-Enhanced Knowledge Graph Representation Learning

[van Benthem et al.,1995] Temporal Logic

[Garcia-Duran et al.,2017a] KBlrn: End-to-End Learning of Knowledge Base Representations with Latent, Relational and Numerical Features

[Garcia-Duran et al.,2017b] TransRev: Modeling Reviews as Translations from Users to Items

[Garcia-Duran et al.,2018] Learning Sequence Encoders for Temporal Knowledge Graph Completion

[Gardner et al.,2015] Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction

[Guo et al.,2016] Jointly Embedding Knowledge Graphs and Logical Rules

[Guu et al.,2016] Traversing the Knowledge Graph in Vector Space.

[Hamilton et al.,2018] Querying Complex Networks in Vector Space

[He et al.,2017] Translation-Based Recommendation

[Hinton,2000] Training Product of Experts by Minimizing Contrastive Divergence

[Jiang et al.,2016] Encoding Temporal Information for Time-Aware Link Prediction

[Kadlec et al.,2017] Knowledge Base Completion: Baselines Strikes Back

References

[Krishna et al.,2015] Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations

[Leblay et al.,2018] Deriving Time Validity in Knowledge Graph

[Onoro-Rubio et al.,2017] Representation Learning for Visual-Relational Knowledge Graphs

[Lonij et al.,2017] Open-World Visual Recognition Using Knowledge Graphs

[Minervini et al.,2017] Adversarial Sets for Regularized Neural Link Predictors

[Neelankatan et al.,2015] Compositional Vector Space Models for Knowledge Base Completion

[Nickel et al.,2017] Poincare Embeddings for Learning Hierarchical Representations

[Rocktaschel et al.,2015] Injecting Logical Background Knowledge into Embeddings for Relation Extraction

[Serban et al.,2016] Generating Factoid Questions with RNN

[Trivedi et al.,2017] Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs

[Wang et al.,2016] Text-Enhanced Representation Learning for Knowledge Graph

[Zitnik et al.,2018] Modeling Polypharmacy Side Effects with Graph Convolutional Networks