ConvE

Convolutional 2D knowledge graph embeddings.

ONE Knowledge graph embedding

TWO Algorithm: ConvE

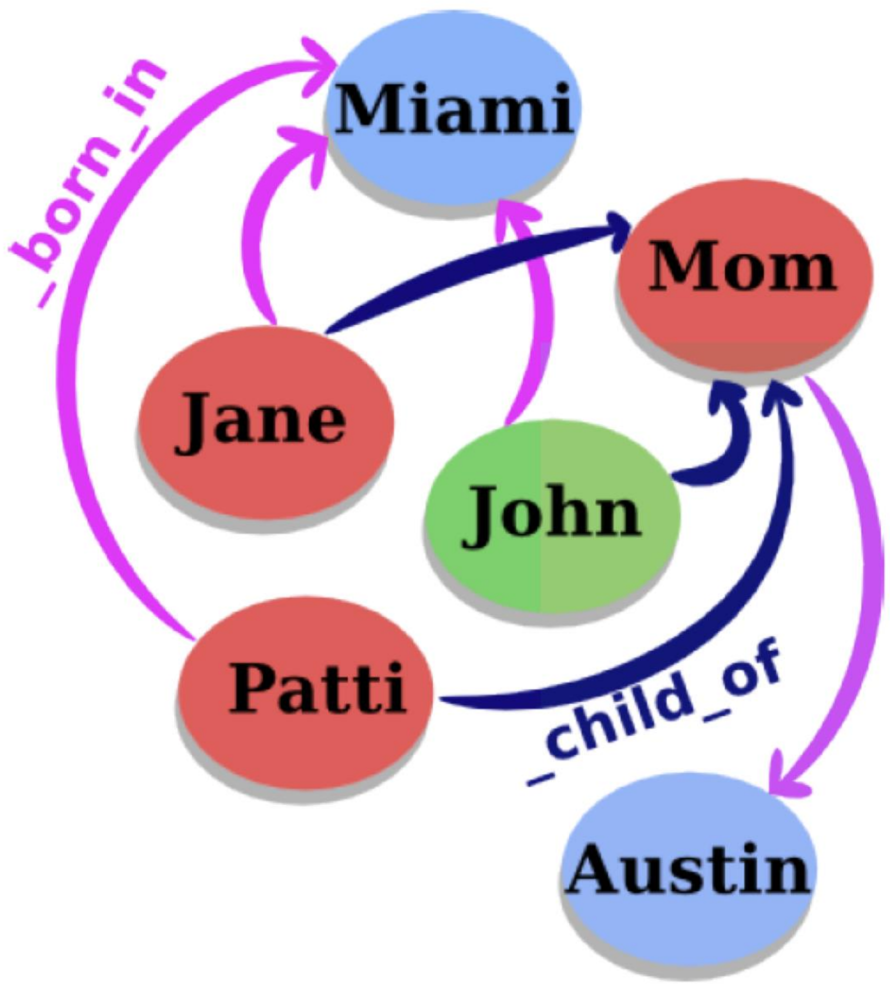
THREE Experiment

PART 1

Knowledge graph embedding

- Knowledge graph
- RL for KG
- Models
- Evaluation: Link prediction

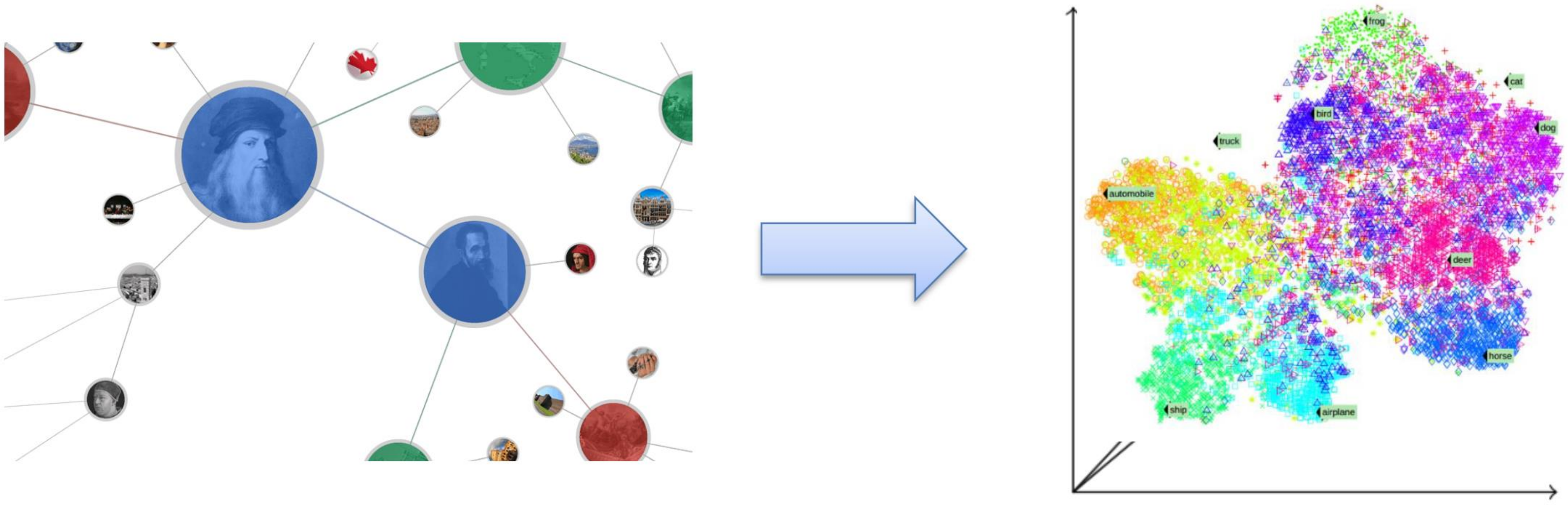
Knowledge graph



- Knowledge structured as graph :
  – Each node = an entity
  – Each edge = a relation
- Fact: (head,relation,tail)
  – Head=subject entity
  – Relation=relation type
  – Tail=object entity
- Typical KGs:
  – WordNet: Lingustic KG
  – Freebase: Word KG

RL for KG

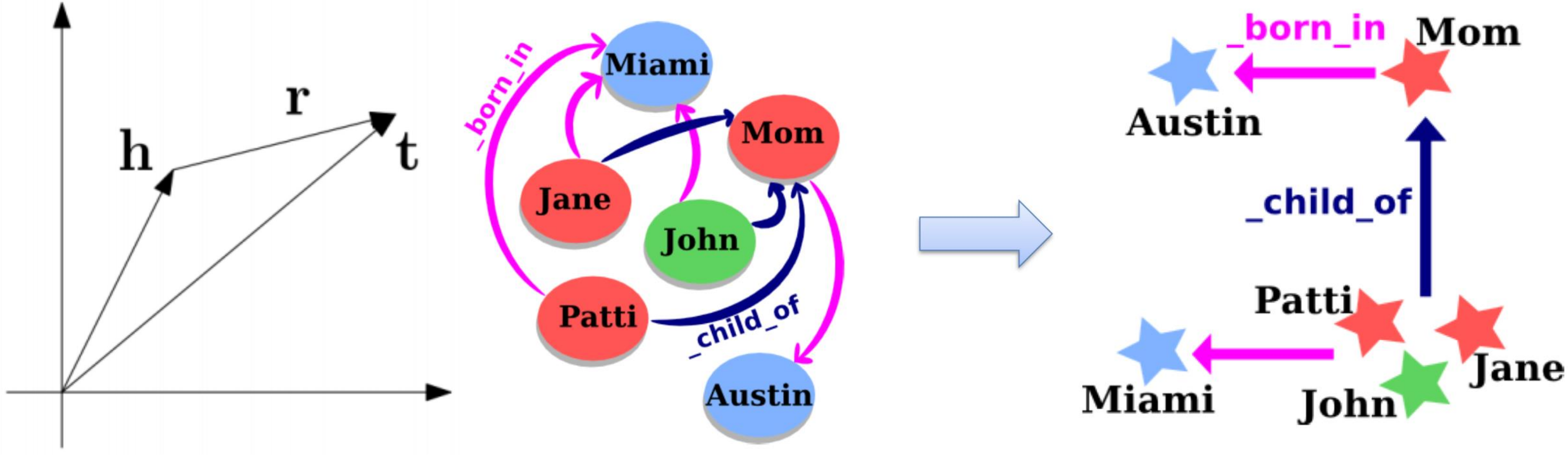
- Typical representations for KG
  - – Symbolic triples (RDF)
  - – Cannot efficiently measure semantic relatedness of entities
- How: Encode KGs into low-dimensional vector spaces



TransE

- For each triple (head, relation, tail), relation as a translation from head to tail
- Learning objective: h + r = t

$$f_r(h,t) = |\boldsymbol{l}_h + \boldsymbol{l}_r - \boldsymbol{l}_t|_{L_1/L_2}$$

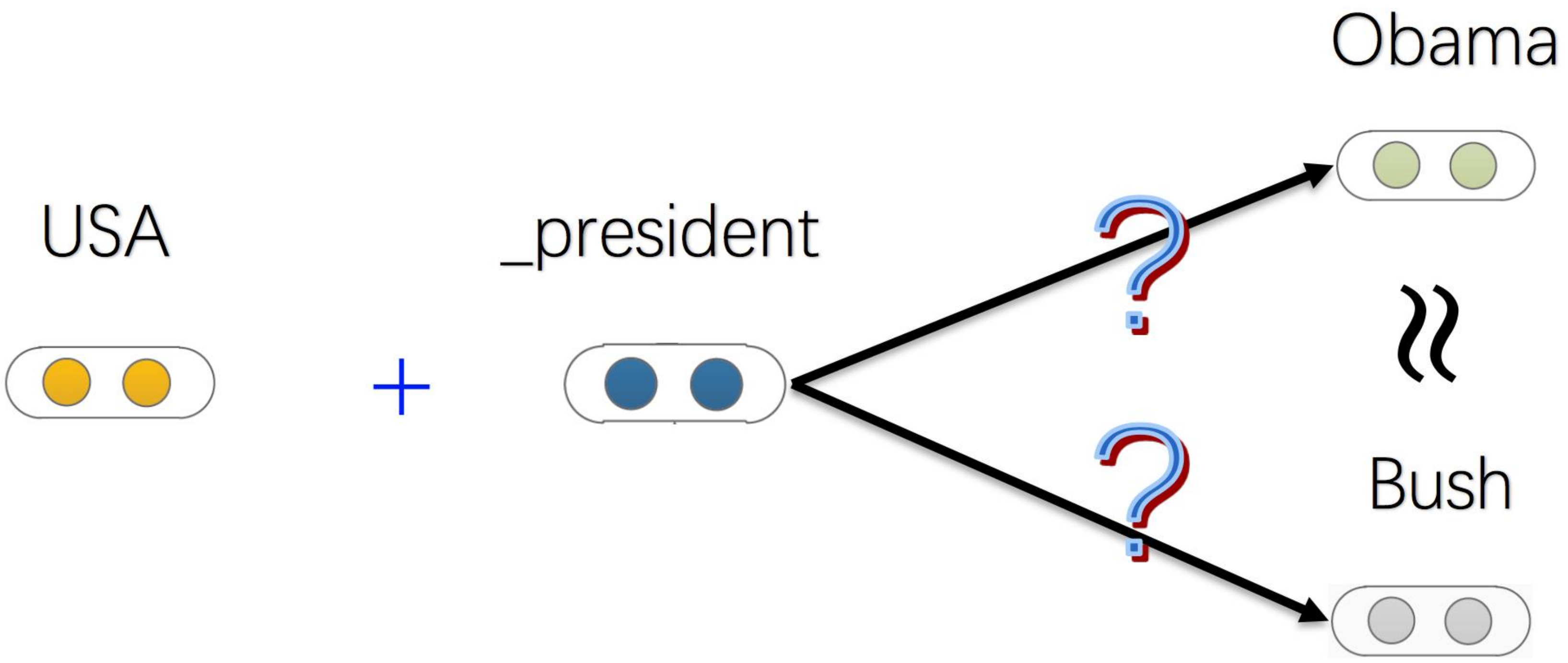


Complex Relations

- 1-to-N, N-to-1, N-to-N relations

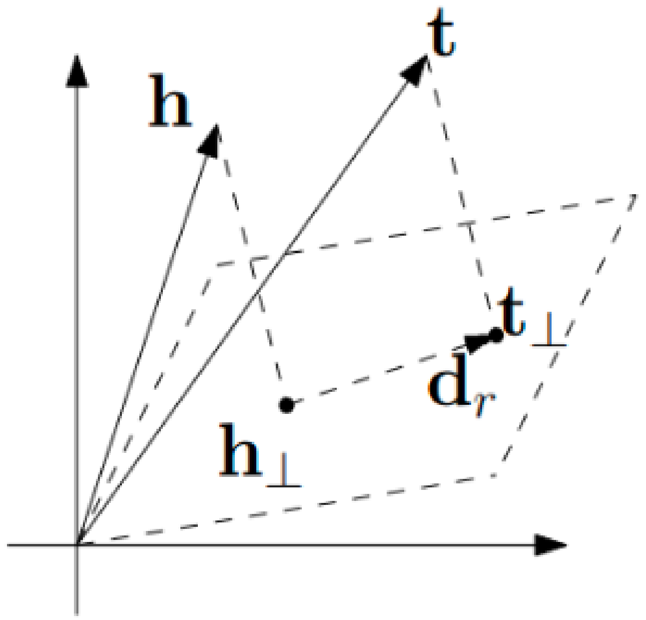
– (USA, _president, Obama)

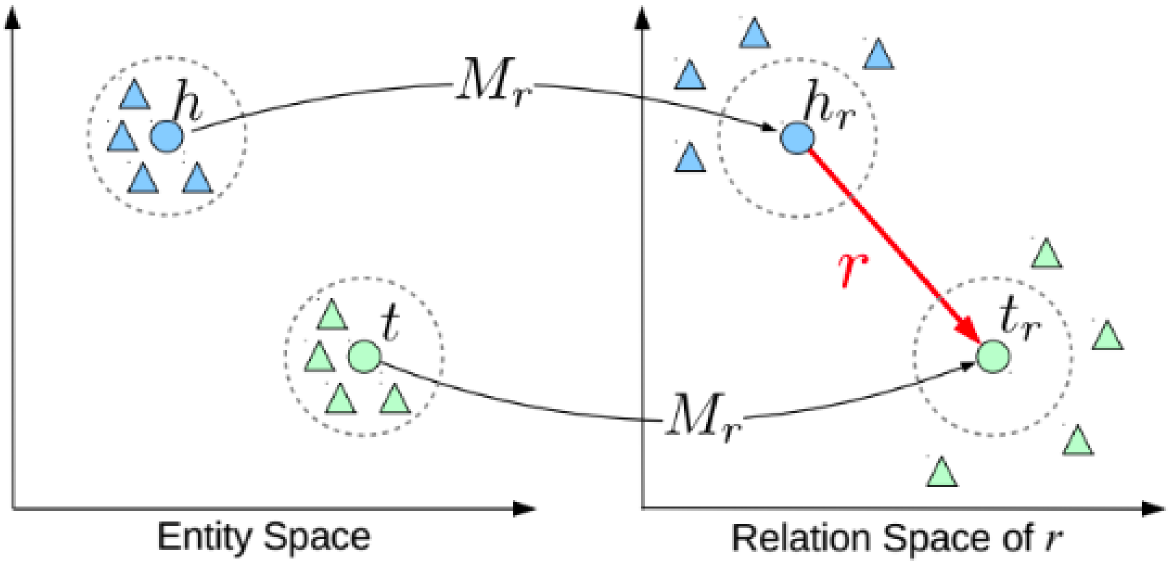
– (USA, _president, Bush)



Complex Relations-based models

Build relation-specific entity embeddings





TransH: $f_r(h,t) = \| \boldsymbol{l}_{h_r} + \boldsymbol{l}_r - \boldsymbol{l}_{t_r} \|_{L_1/L_2}$

TransR: $f_r(h,t) = \| \boldsymbol{l}_{h_r} + \boldsymbol{l}_r - \boldsymbol{l}_{t_r} \|_{L_1/L_2}$

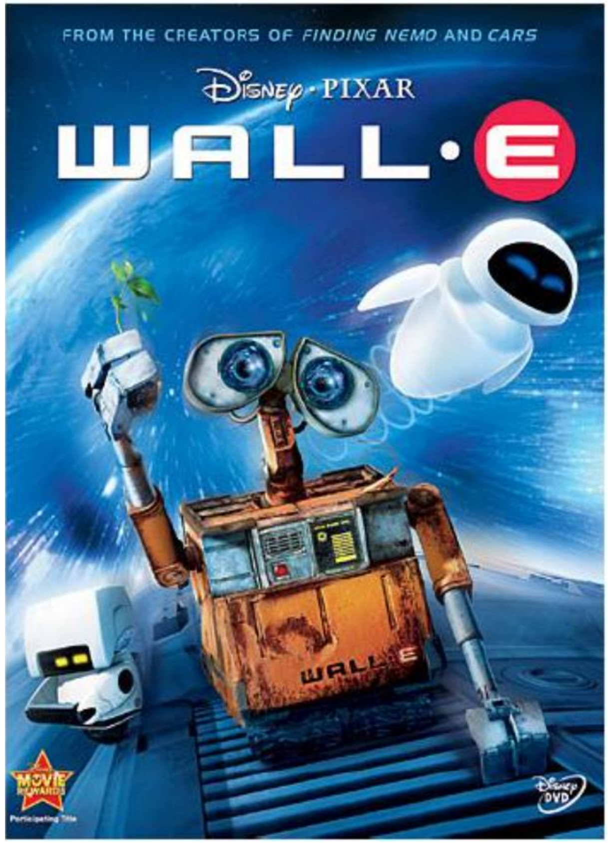
Wang, et al. (2014). Knowledge graph embedding by translating on hyperplanes. AAAI.
Lin, et al. (2015). Learning entity and relation embeddings for knowledge graph completion. AAAI.

More Models

- NTN : models KG with a Neural Tensor Network and represents entities via word vectors
- Holographic Embeddings (Hole) :uses the circular correlation to represent entity pairs
- DistMult :embeddings learned from the bilinear objective are particularly good at capturing relationa l semantics and that the composition of relations is characterized by matrix multiplication
- ComplEx : factorizing the relational tensor using a logistic or hinge loss
- R-GCN ： GCN provides a framework for representing learning for graph nodes. R-gcn provides a triple self-encoding and decoding scoring method;

Evaluation: Link prediction

WALL-E



_has_genre

Animation

Computer animation

Comedy film

Adventure film

Science Fiction

Fantasy

Stop motion

Satire

Drama

Connecting

PART 2

Algorithm: ConvE

- The main idea
- ConvE Algorithm
- Fast Evaluation

Main idea

problems:

- HolE does not learn multiple layers of non-linear features

methods:

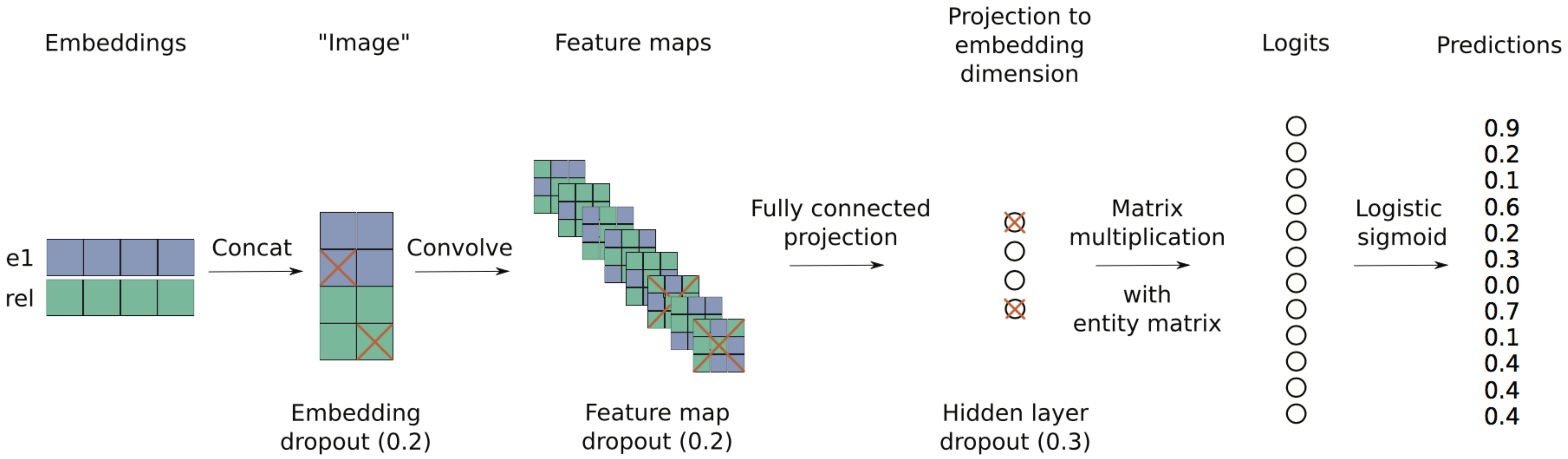
- Using 2D rather than 1D convolutions increases the expressiveness through additional points of interaction between embeddings

$$([a \quad a \quad a]\,;[b \quad b \quad b]) = [a \quad a \quad a \quad b \quad b \quad b]\,.$$

$$\left(\begin{bmatrix} a & a & a \\ a & a & a \end{bmatrix};\begin{bmatrix} b & b & b \\ b & b & b \end{bmatrix}\right) = \begin{bmatrix} a & a & a \\ a & a & a \\ b & b & b \\ b & b & b \end{bmatrix} \longrightarrow \begin{bmatrix} a & a & a \\ b & b & b \\ a & a & a \\ b & b & b \end{bmatrix}$$

ConvE Algorithm

Scoring function: $$\psi_r(\mathbf{e}_s, \mathbf{e}_o) = f(\mathrm{vec}(f([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_r}] * \omega))\mathbf{W})\mathbf{e}_o,$$



Fast Evaluation

- For link prediction models, the batch size is usually increased to speed up evaluation
- **Problem** : It is not feasible for convolutional models
  - the memory requirements quickly outgrow the GPU memory capacity
- **Evaluation**:
  - 1-1 scoring : take an entity pair and a relation as a triple (s, r, o), and score it
  - 1-N scoring : take one (s, r) pair and score it against all entities simultaneously

Eg:	A	x	B	entities:4	relations:2
	C	x	A	1-1 scoring:	4*2*4=32
	D	y	C	1-N scoring:	3*4= 12

PART 3

Experiment

- Inverse model
- Datasets
- Evaluation
- Results & Analysis

Inverse model

- Test leakage : a large number of test triples can be obtained simply by inverting triples in the training set.
  - Eg: (s, hypernym, o) ——(o, hyponym , s)
  - WN18 and FB15k have 94% and 81% test leakage as inverse relations
- Inverse model: solely models inverse relations
  - Training: given two relation pairs r1,r2 ∈R, check whether (s,r1,o) implies (o, r2 , s), or vice-versa.
  - Testing: check if the test triple has inverse matches outside the test set

Datasets

Datasets	Entities	Relations	Subset of	Description
WN18	40943	18	WordNet	hyponym and hypernym relations
FB15k	14951	1345	Freebase	facts about movies, actors, awards, sports, and sport teams
YAGO3-10	123181	37	YAGO3	attributes of people, such as citizenship, gender, and profession
Countries	\	\	\	three sub-tasks which increase in difficulty to evaluate a model’s ability to learn long-range dependencies between entities and relations.
WN18RR	40943	11	WN18	inverse relations are removed (this paper)
FB15k-237	14541	237	FB15k	inverse relations are removed (Toutanova and Chen (2015))

Evaluation

Evaluation	Description
Hits@k	The percentage of the first k results contain the correct entity
MR	The average ranking
MRR	Mean Reciprocal Rank
AUC-PR	\

Query	Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, **cats**	cats	3	1/3
torus	torii, **tori**, toruses	tori	2	1/2
virus	**viruses**, virii, viri	viruses	1	1

MRR值为：(1/3 + 1/2 + 1)/3 = 11/18=0.61

Results

Table 3: Link prediction results for WN18 and FB15k

	WN18					FB15k				
			Hits					Hits		
	MR	MRR	@10	@3	@1	MR	MRR	@10	@3	@1
DistMult (Yang et al. 2015)	902	.822	.936	.914	.728	97	.654	.824	.733	.546
ComplEx (Trouillon et al. 2016)	–	.941	.947	.936	.936	–	.692	.840	.759	.599
Gaifman (Niepert 2016)	**352**	–	.939	–	.761	75	–	.842	–	.692
ANALOGY (Liu, Wu, and Yang 2017)	–	**.942**	.947	.944	**.939**	–	.725	.854	.785	.646
R-GCN (Schlichtkrull et al. 2017)	–	.814	.964	.929	.697	–	.696	.842	.760	.601
ConvE	504	**.942**	.955	.947	.935	**64**	**.745**	**.873**	**.801**	.670
Inverse Model	567	.861	**.969**	**.968**	.764	1897	.706	.737	.718	**.689**

Table 4: Link prediction results for WN18RR and FB15k-237

	WN18RR					FB15k-237				
			Hits					Hits		
	MR	MRR	@10	@3	@1	MR	MRR	@10	@3	@1
DistMult (Yang et al. 2015)	**5110**	.43	.49	.44	.39	254	.241	.419	.263	.155
ComplEx (Trouillon et al. 2016)	5261	.44	**.51**	**.46**	**.41**	339	.247	.428	.275	.158
R-GCN (Schlichtkrull et al. 2017)	–	–	–	–	–	–	.248	.417	.258	.153
ConvE	5277	**.46**	.48	.43	.39	**246**	**.316**	**.491**	**.350**	**.239**
Inverse Model	13219	.36	.36	.36	.36	7148	.009	.012	.010	.006

Results

Table 5: Link prediction results for YAGO3-10 and Countries

	YAGO3-10					**Countries**		
			Hits			AUC-PR		
	MR	MRR	@10	@3	@1	S1	S2	S3
DistMult (Yang et al. 2015)	5926	.34	.54	.38	.24	**1.00±0.00**	0.72±0.12	0.52±0.07
ComplEx (Trouillon et al. 2016)	6351	.36	.55	.40	.26	0.97±0.02	0.57±0.10	0.43±0.07
ConvE	**2792**	**.52**	**.66**	**.56**	**.45**	**1.00±0.00**	**0.99±0.01**	**0.86 ±0.05**
Inverse Model	60251	.02	.02	.02	.01	–	–	–

Analysis

Table 6: Mean PageRank $\times 10^{-3}$ of nodes in the test set vs reduction in error in terms of AUC-PR or Hits@10 of ConvE wrt. DistMult.

Dataset	PageRank	Error Reduction
WN18RR	0.104	0.91
WN18	0.125	1.28
FB15k	0.599	1.23
FB15-237	0.733	1.17
YAGO3-10	0.988	1.91
Countries S3	1.415	3.36
Countries S1	1.711	0.00
Countries S2	1.796	18.6

Table 7: Ablation study for FB15k-237.

Ablation	Hits@10
Full ConvE	0.491
Hidden dropout	-0.044 $\pm$ 0.003
Input dropout	-0.022 $\pm$ 0.000
1-N scoring	-0.019
Feature map dropout	-0.013 $\pm$ 0.001
Label smoothing	-0.008 $\pm$ 0.000

hypothesis : deeper models have an advantage to model more complex graphs.

model nodes with high indegree with greater precision – which is possibly related to its depth.

END

THANK YOU