Part II: Models based on extra information

TOPICS:

- INCORPORATING ENTITY TYPES
- INCORPORATING TEXTUAL DESCRIPTIONS
- INCORPORATING RELATION PATHS
- INCORPORATING LOGIC RULES

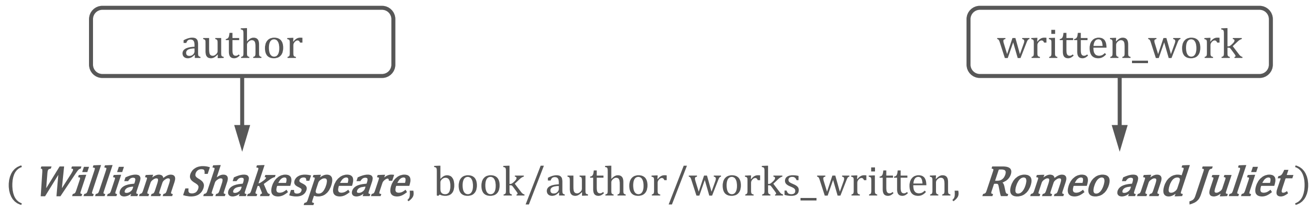
Part II: Models based on extra information

TOPICS:

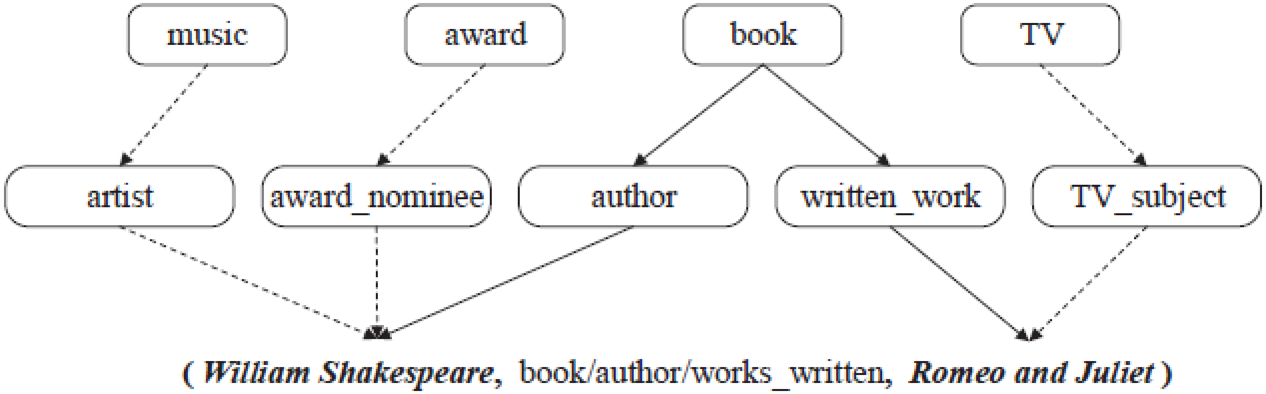
- INCORPORATING ENTITY TYPES
- INCORPORATING TEXTUAL DESCRIPTIONS
- INCORPORATING RELATION PATHS
- INCORPORATING LOGIC RULES

Entity types

- Semantic categories to which entities belong



- Each entity may have multiple type labels, and the types could also be hierarchical



Example from: Xie et al. (2016). Representation learning of knowledge graphs with hierarchical types. IJCAI'16.

Semantically smooth embedding (Guo et al., 2015)

- Key idea
  - Entities of the same type should stay close in the embedding space
- Modeling semantic smoothness
  - Laplacian eigenmaps (LE): If two entities belong to the same type, they will have their embeddings similar to each other

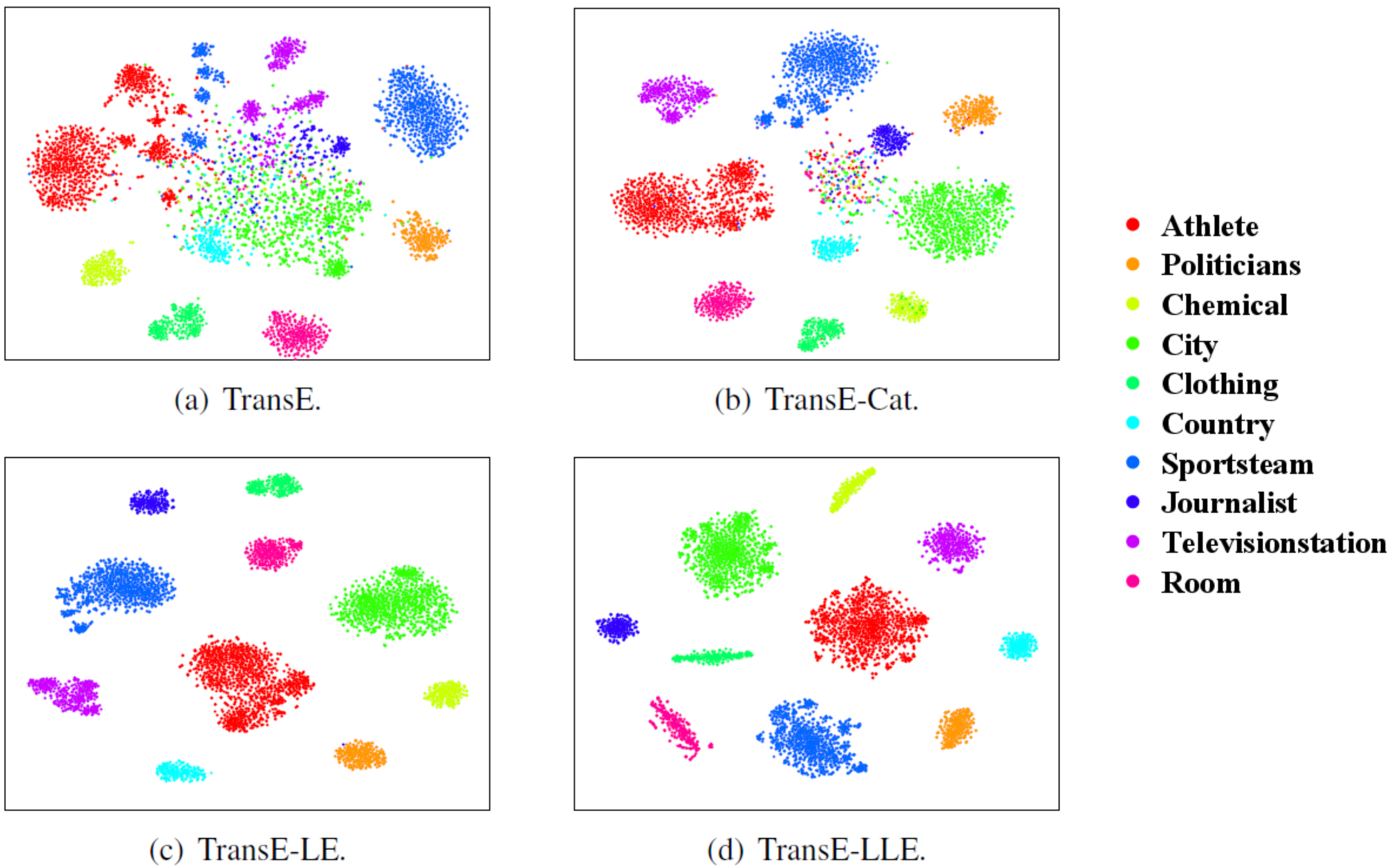
$$\mathcal{R}_1 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{e}_i - \mathbf{e}_j\|^2 w_{ij}^1, \quad w_{ij}^1 = \begin{cases} 1, & type(e_i) = type(e_j) \\ 0, & \text{otherwise} \end{cases}$$

  - Locally linear embedding (LLE): An entity can be reconstructed from its near neighbors (entities of the same type) in the embedding space

$$\mathcal{R}_2 = \sum_{i=1}^{n}\|\mathbf{e}_i - \sum_{e_j \in \mathcal{N}(e_i)} w_{ij}^2 \mathbf{e}_j\|^2, \quad w_{ij}^2 = \begin{cases} \frac{1}{|\mathcal{N}(e_i)|}, & e_j \in \mathcal{N}(e_i) \\ 0, & \text{otherwise} \end{cases}$$

Semantically smooth embedding (cont.)

- Visualization of entity vectors learned by semantically smooth embedding (Guo et al., 2015)



Type-embodied knowledge representation learning (Xie et al., 2016b)

Key idea

- Translation after type-specific entity projection: $\mathbf{M}_{rs}\mathbf{s} + \mathbf{r} \approx \mathbf{M}_{ro}\mathbf{o}$

Modeling multiple type labels

- Projecting an entity with a linear combination of type matrices

$$\mathbf{M}_{rs} = \frac{\sum_{i=1}^{n_s} \alpha_i \mathbf{M}_{c_i}}{\sum_{i=1}^{n_s} \alpha_i}, \quad \alpha_i = \begin{cases} 1, & c_i \in C_{r_s} \\ 0, & c_i \notin C_{r_s} \end{cases}$$

projection matrix of type c_i

Modeling hierarchical types

- Projection matrix of a type as a composition of projection matrices of its sub-types

$$\text{addition:} \quad \mathbf{M}_{c_i} = \beta_1 \mathbf{M}_{c_i^1} + \cdots + \beta_\ell \mathbf{M}_{c_i^\ell}$$

$$\text{multiplication:} \quad \mathbf{M}_{c_i} = \mathbf{M}_{c_i^1} \odot \cdots \odot \mathbf{M}_{c_i^\ell}$$

projection matrix of sub-type c_i^1

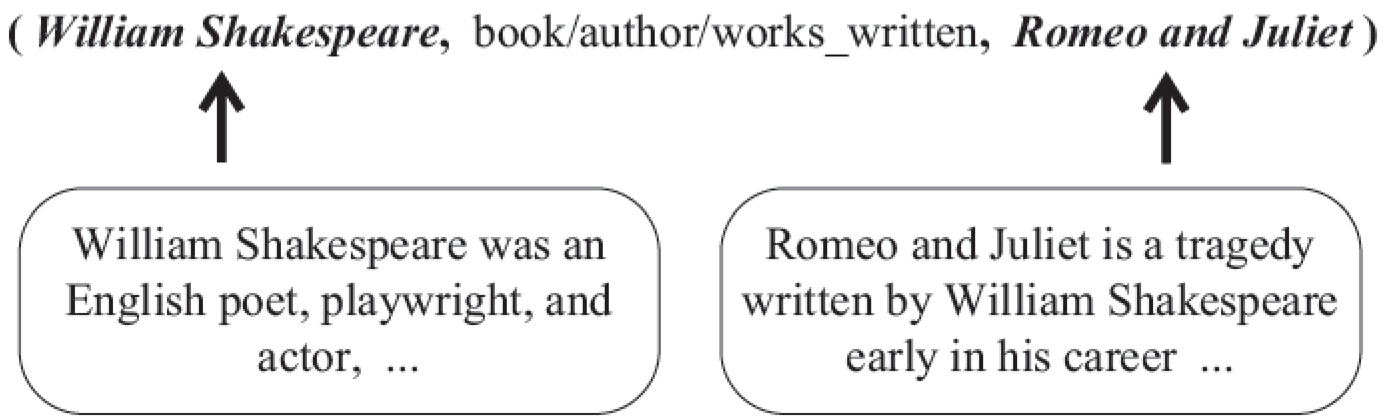
Part II: Models based on extra information

TOPICS:

- INCORPORATING ENTITY TYPES
- <u>INCORPORATING TEXTUAL DESCRIPTIONS</u>
- INCORPORATING RELATION PATHS
- INCORPORATING LOGIC RULES

Textual descriptions

- Concise descriptions of entities in knowledge graphs

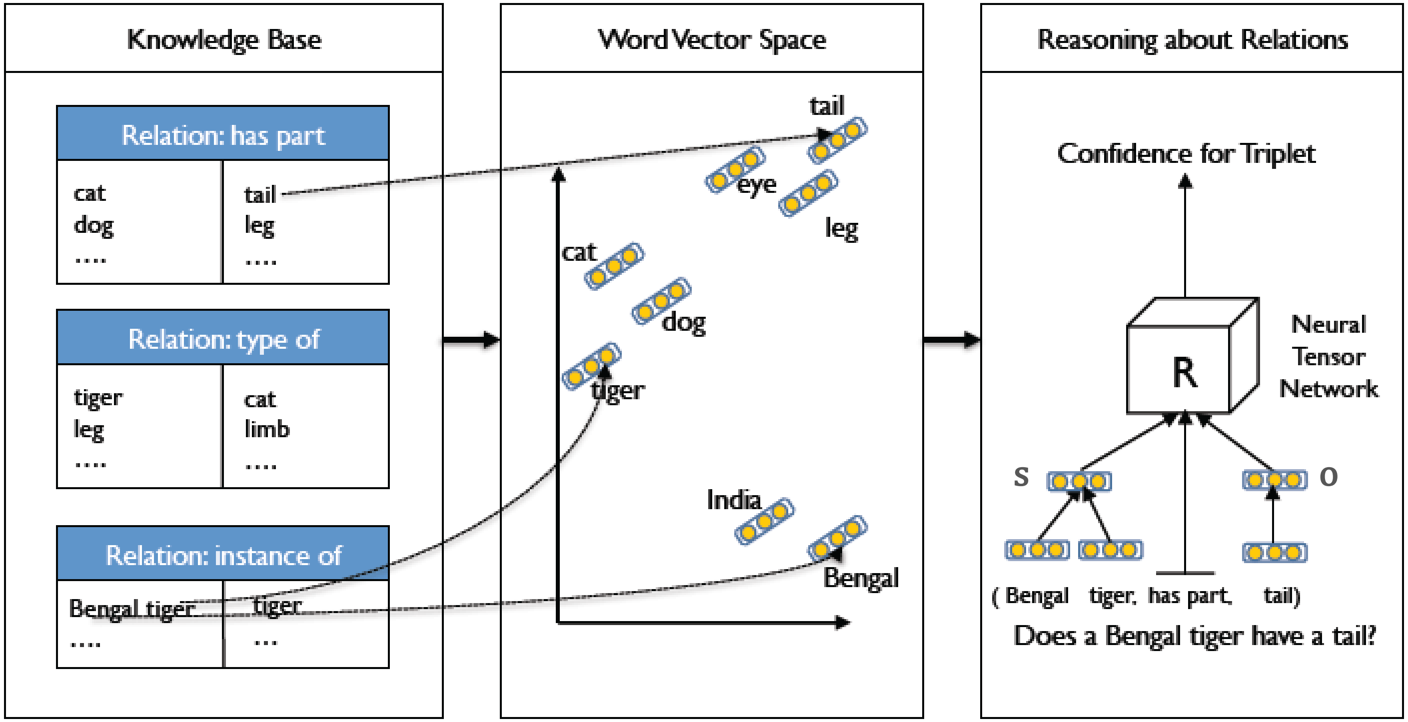


- Other general textual information such as news releases and Wikipedia articles

Example from: Xie et al. (2016). Representation learning of knowledge graphs with entity descriptions. AAAI'16.

Initialization by word embeddings (Socher et al., 2013)

- Key idea
  - Initializing entity representations with pre-trained word embeddings



$$\text{vec}(\text{Bengal tiger}) = \frac{1}{2}\big(\text{vec}(\text{Bengal}) + \text{vec}(\text{tiger})\big)$$

Jointly embedding with text data (Wang et al., 2014; Zhong et al., 2015)

- **Key idea**
  - Jointly embedding relations, entities, and words into the same vector space so that one can make predictions between entities and words
- **Jointly embedding framework**

$$\min \quad \mathcal{L}_K + \mathcal{L}_T + \mathcal{L}_A$$

knowledge model — text model — alignment model

  - Knowledge model: Modeling triples in a knowledge graph
  - Text model: Modeling co-occurring word pairs in a text corpus
  - Alignment model: Aligning entity and word embedding spaces
    - Wikipedia anchors, entity names, entity descriptions

Jointly embedding with text data (cont.)

Knowledge model

$$\Pr(s|r,o) = \frac{\exp\{z(s,r,o)\}}{\sum_{s' \in \mathcal{E}} \exp\{z(s',r,o)\}} \quad z(s,r,o) = b - 0.5\|\mathbf{s} + \mathbf{r} - \mathbf{o}\|$$

$$\mathcal{L}_K = -\sum_{(s,r,o)\in\mathcal{T}^+} [\log \Pr(s|r,o) + \log \Pr(r|s,o) + \log \Pr(o|s,r)]$$

Text model

$$\Pr(w|v) = \frac{\exp\{z(w,v)\}}{\sum_{w' \in \mathcal{V}} \exp\{z(w',v)\}} \quad z(w,v) = b - 0.5\|\mathbf{w} - \mathbf{v}\|$$

$$\mathcal{L}_T = -\sum_{(w,v)\in\mathcal{C}} \log \Pr(w|v)$$

Alignment model (by entity descriptions)

$$\Pr(w|e) = \frac{\exp\{z(e,w)\}}{\sum_{w' \in \mathcal{V}} \exp\{z(e,w')\}} \quad z(e,w) = b - 0.5\|\mathbf{e} - \mathbf{w}\|$$

$$\mathcal{L}_A = -\sum_{e\in\mathcal{E}} \sum_{w\in\mathcal{D}_e} [\log \Pr(w|e) + \log \Pr(e|w)]$$

Description-Embodied Knowledge Representation Learning (Xie et al., 2016a)

Key idea

- Entity: structure-based embedding + description-based embedding
- Description-based embedding as composition of word embeddings

Triple scoring function

$$f(s,r,o) = -\|\mathbf{s}_K + \mathbf{r} - \mathbf{o}_K\| \quad \text{① score for structure –based embeddings}$$

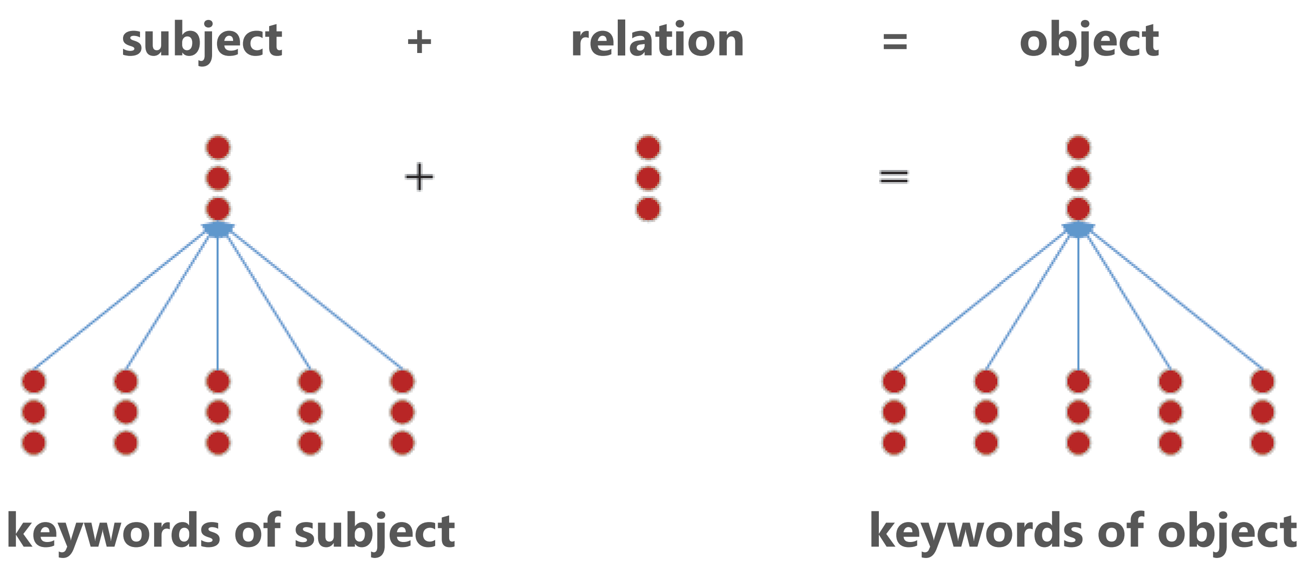
$$-\|\mathbf{s}_T + \mathbf{r} - \mathbf{o}_T\| - \|\mathbf{s}_K + \mathbf{r} - \mathbf{o}_T\| - \|\mathbf{s}_T + \mathbf{r} - \mathbf{o}_K\|$$

② score for description –based embeddings

- $\mathbf{s}_K, \mathbf{o}_K$: structure-based entity embeddings
- $\mathbf{s}_T, \mathbf{o}_T$: description-based entity embeddings, modeled as compositions of word embeddings

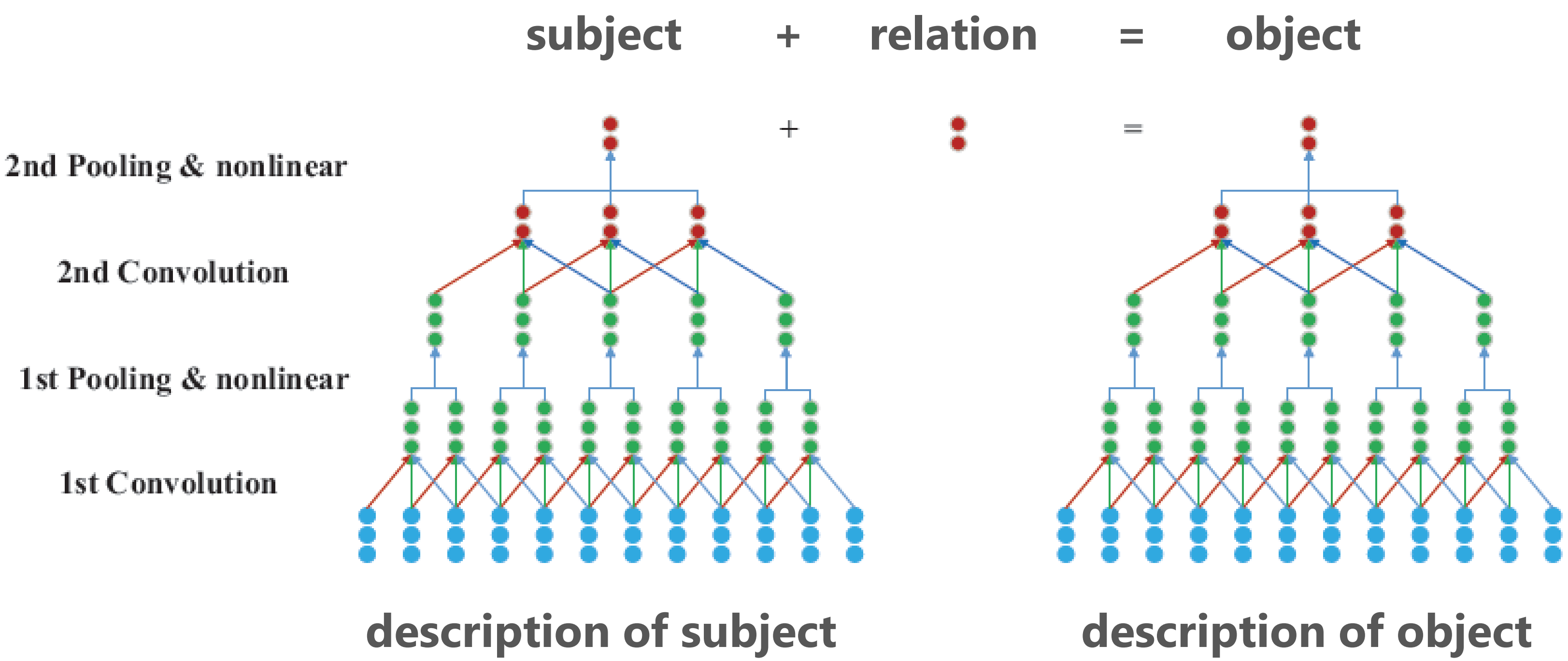
Description-Embodied Knowledge Representation Learning (cont.)

- Modeling description-based entity embeddings
  - Continuous bag-of-words encoder: Composition by addition, ignoring word orders



Description-Embodied Knowledge Representation Learning (cont.)

- Modeling description-based entity embeddings
  - Convolutional neural network encoder: Composition by CNN, taking word orders into account



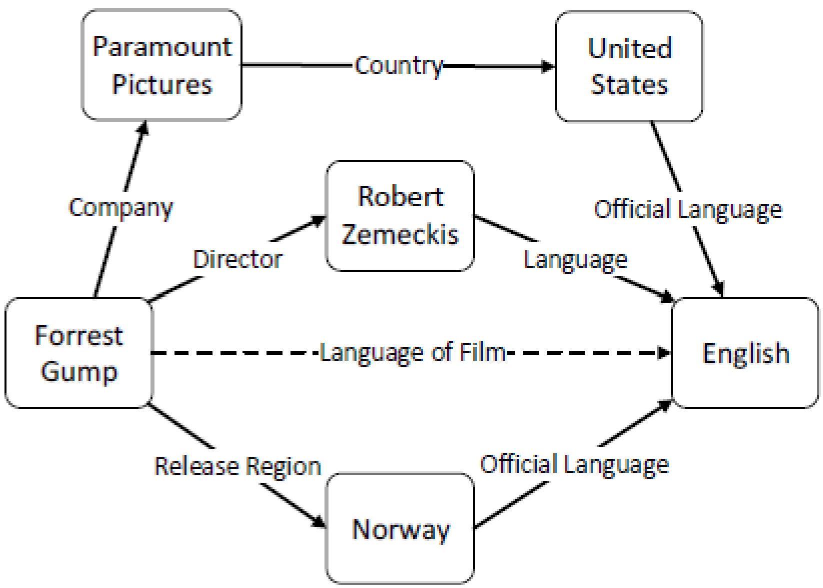
Part II: Models based on extra information

TOPICS:

- INCORPORATING ENTITY TYPES
- INCORPORATING TEXTUAL DESCRIPTIONS
- INCORPORATING RELATION PATHS
- INCORPORATING LOGIC RULES

Relation paths

- Multi-hop relationships between entities, extremely useful for predicting missing links



$\text{Forrest Gump} \xrightarrow{\text{Company}} \text{Paramount Pictures} \xrightarrow{\text{Country}} \text{United States} \xrightarrow{\text{Official Language}} \text{English}$

$\text{Forrest Gump} \xrightarrow{\text{Director}} \text{Robert Zemeckis} \xrightarrow{\text{Language}} \text{English}$

$\text{Forrest Gump} \xrightarrow{\text{Release Region}} \text{Norway} \xrightarrow{\text{Official Language}} \text{English}$

➡ $\text{Forrest Gump} \overset{?}{\xrightarrow{\text{Language of Film}}} \text{English}$

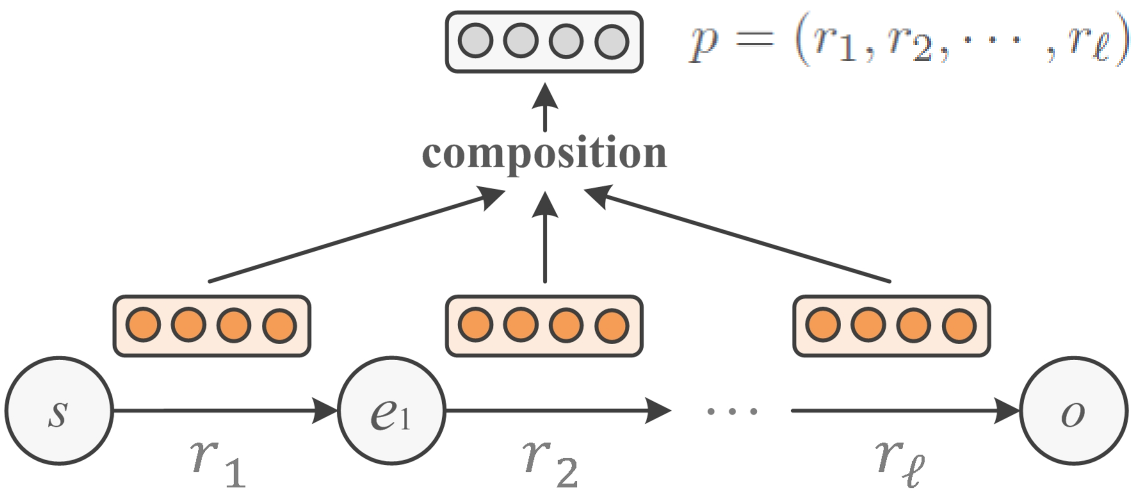
Example from: Lin et al. (2015). Modeling relation paths for representation learning of knowledge bases. EMNLP’15.

Modeling relation paths

- Relation path: A sequence of relations linking two entities

$$p = (r_1, r_2, \cdots, r_\ell) \quad \Leftrightarrow \quad s \xrightarrow{r_1} e_1 \xrightarrow{r_2} \cdots \xrightarrow{r_\ell} o$$

- Path representation: Composition of relation representations

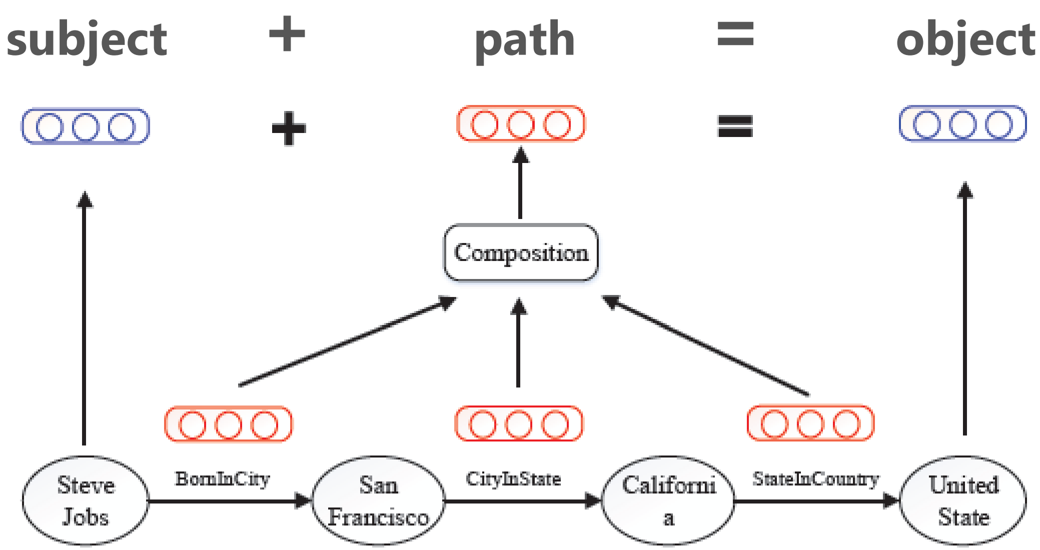


$$\mathbf{p} = \mathbf{r}_1 \circ \mathbf{r}_2 \circ \cdots \circ \mathbf{r}_\ell \quad (\circ \text{ is a composition operation})$$

Path-based TransE (Lin et al., 2015)

Key idea

- Taking relation paths as translations between long distance entities



Semantic composition

$$\text{addition:} \quad \mathbf{p} = \mathbf{r}_1 + \mathbf{r}_2 + \cdots + \mathbf{r}_\ell$$

$$\text{multiplication:} \quad \mathbf{p} = \mathbf{r}_1 \odot \mathbf{r}_2 \odot \cdots \odot \mathbf{r}_\ell$$

$$\text{RNN:} \quad \mathbf{c}_i = f\left(\mathbf{W}[\mathbf{c}_{i-1}; \mathbf{r}_i]\right)$$

Path-based TransE (cont.)

- Optimization problem
  - Modeling relation-connected triple (s, r, o)

$$\mathcal{L}(s,r,o) = \sum_{(s',r',o')\in\mathcal{T}^-} \max\left(0, \gamma + \|\mathbf{s}+\mathbf{r}-\mathbf{o}\| - \|\mathbf{s}'+\mathbf{r}'-\mathbf{o}'\|\right)$$

loss of triple

  - Modeling path-connected triple (s, p, o)

$$f(s,p,o) = -\|\mathbf{s}+\mathbf{p}-\mathbf{o}\| = -\|\mathbf{p}-(\mathbf{o}-\mathbf{s})\| \approx -\|\mathbf{p}-\mathbf{r}\|$$

$$\mathcal{L}(p,r) = \sum_{(s,r',o)\in\mathcal{T}^-} \max\left(0, \gamma + \|\mathbf{p}-\mathbf{r}\| - \|\mathbf{p}-\mathbf{r}'\|\right)$$

loss of path

  - Combining the two parts

$$\min \sum_{(s,r,o)\in\mathcal{T}^+} \left[\mathcal{L}(s,r,o) + \frac{1}{Z}\sum_{p\in\mathcal{P}(s,o)} R(p|s,o)\mathcal{L}(p,r)\right]$$

reliability of path p

① loss of (s, r, o) ② loss of all paths linking s and o

Path-based TransE (cont.)

- Link prediction performance on FB15k

Metric	Mean Rank		Hits@10 (%)	
	Raw	Filter	Raw	Filter
RESCAL	828	683	28.4	44.1
SE	273	162	28.8	39.8
SME (linear)	274	154	30.7	40.8
SME (bilinear)	284	158	31.3	41.3
LFM	283	164	26.0	33.1
TransE	243	125	34.9	47.1
TransH	212	87	45.7	64.4
TransR	**198**	77	48.2	68.7
TransE (Our)	205	63	47.9	70.2
PTransE (ADD, 2-step)	200	**54**	**51.8**	83.4
PTransE (MUL, 2-step)	216	67	47.4	77.7
PTransE (RNN, 2-step)	242	92	50.6	82.2
PTransE (ADD, 3-step)	207	58	51.4	**84.6**

$$s \xrightarrow{r_1} e_1 \xrightarrow{r_2} o$$

$$\mathbf{s} + \mathbf{r}_1 = \mathbf{e}_1$$

$$\mathbf{e}_1 + \mathbf{r}_2 = \mathbf{o}$$

⬇

$$\mathbf{s} + (\mathbf{r}_1 + \mathbf{r}_2) = \mathbf{o}$$

Addition composition operation performs best

Compositionalizing other models (Guu et al., 2015)

- TransE (composition by addition)

$$f(s, r, o) = -\|\mathbf{s} + \mathbf{r} - \mathbf{o}\| \Rightarrow f(s, p, o) = -\|\mathbf{s} + (\mathbf{r}_1 + \cdots + \mathbf{r}_\ell) - \mathbf{o}\|$$

- RESCAL (composition by multiplication)

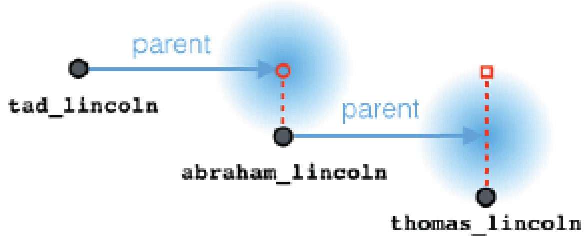
$$f(s, r, o) = \mathbf{s}^\top \mathbf{M}_r \mathbf{o} \Rightarrow f(s, p, o) = \mathbf{s}^\top (\mathbf{M}_{r_1} \odot \cdots \odot \mathbf{M}_{r_\ell}) \mathbf{o}$$

- DistMult (composition by multiplication)

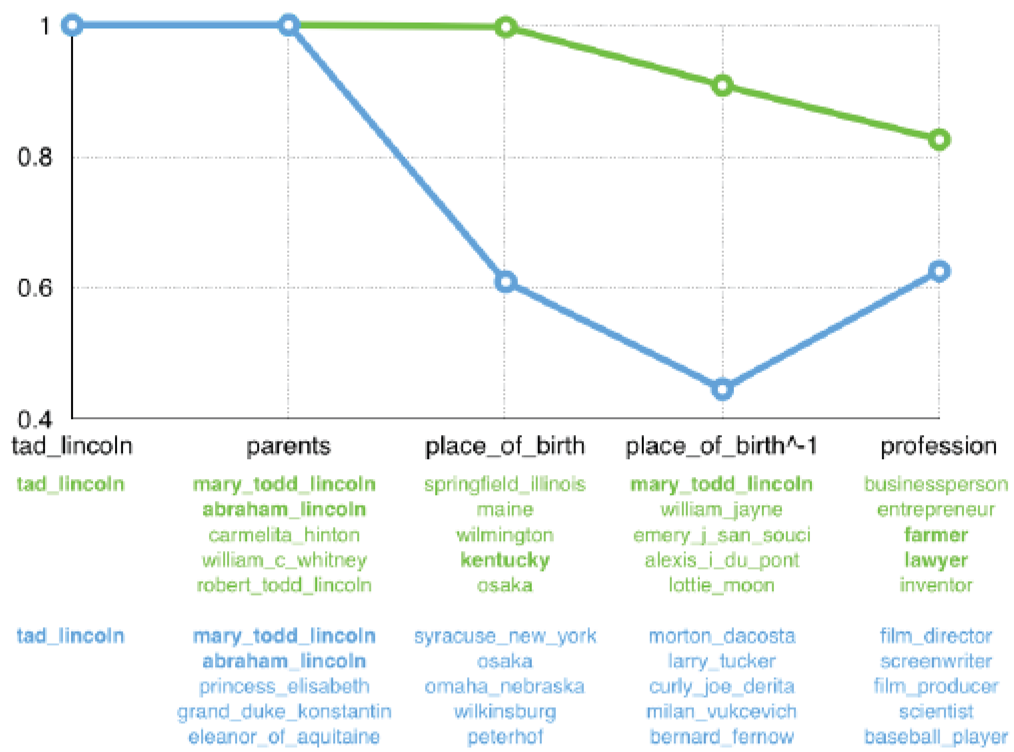
$$f(s, r, o) = \mathbf{s}^\top \mathrm{diag}(\mathbf{r}) \mathbf{o} \Rightarrow f(s, p, o) = \mathbf{s}^\top \mathrm{diag}(\mathbf{r}_1 \odot \cdots \odot \mathbf{r}_\ell) \mathbf{o}$$

Advantage of compositional learning (Guu et al., 2015)

- Modeling triples separately introduces cascading errors



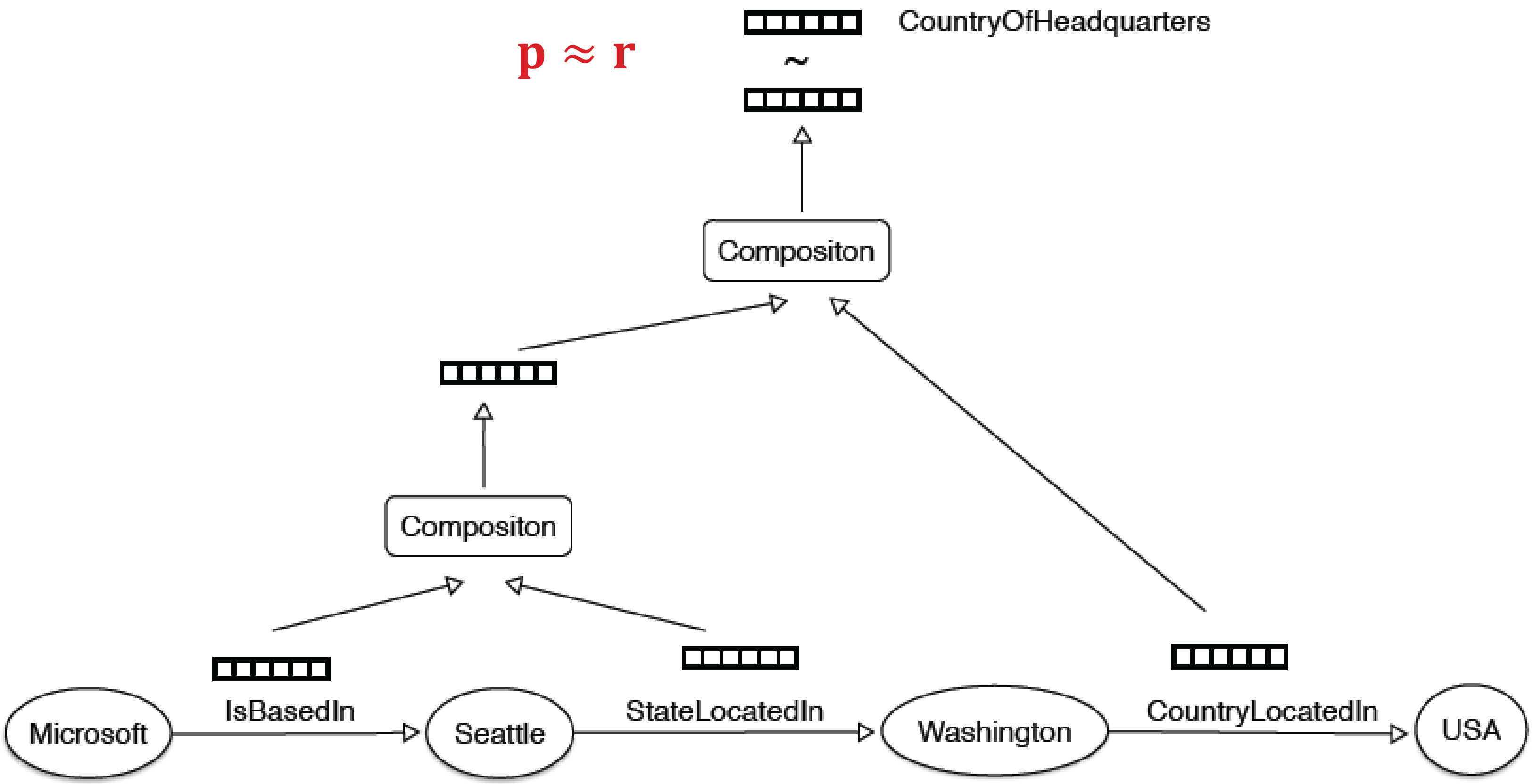
- Modeling paths compositionally reduces cascading errors



Path-RNN (Neelakantan et al., 2015)

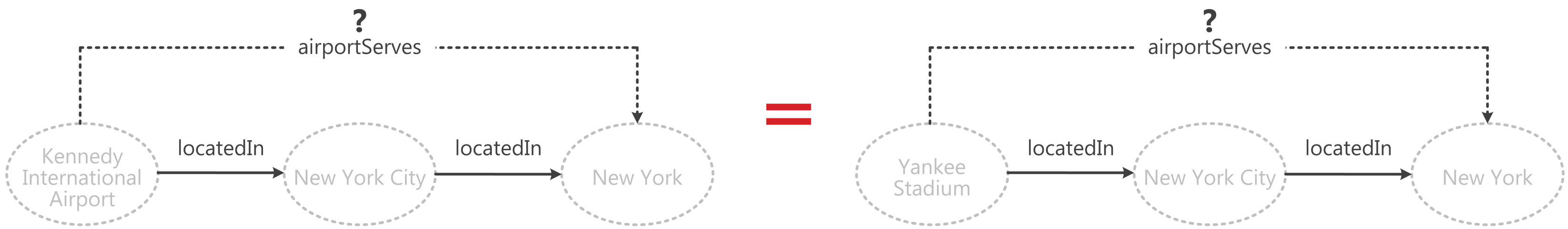
- Modeling paths (sequences of relations) with RNN

$\mathbf{p} \approx \mathbf{r}$

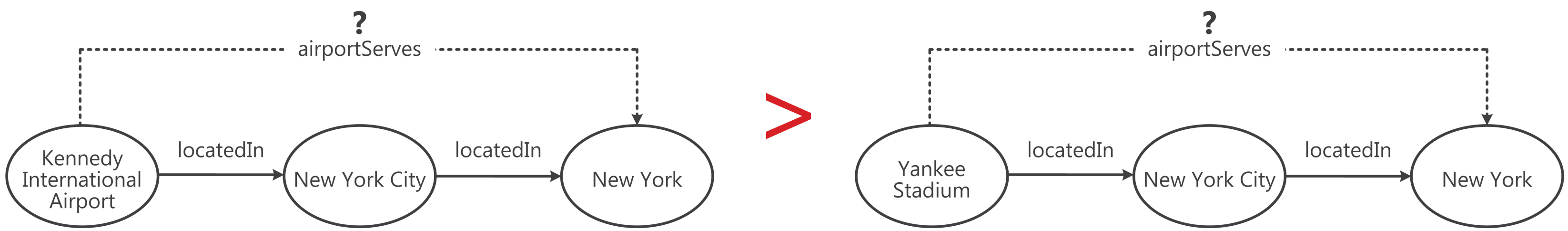


Path-RNN with entity information (Das et al., 2017)

- Entity information is useful in path modeling
  - Without entity information

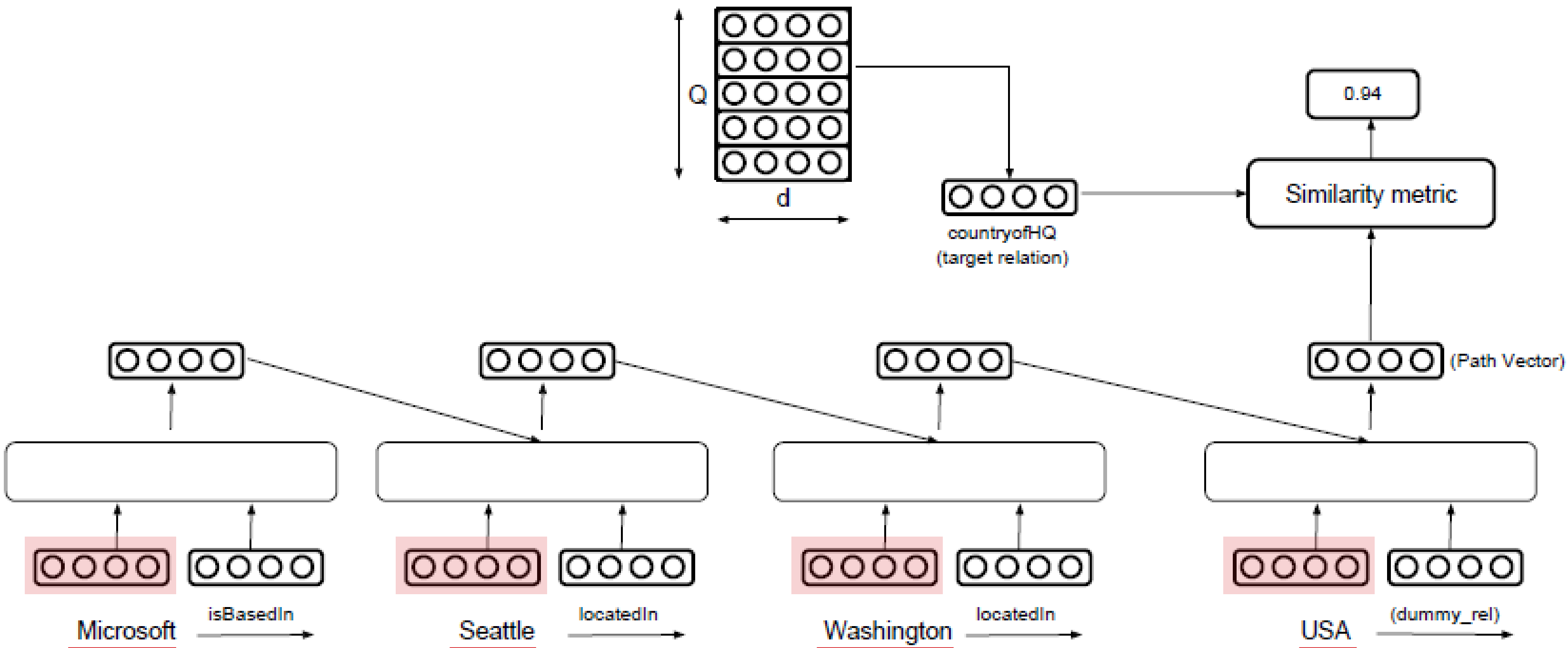


  - With entity information



Path-RNN with entity information (cont.)

- Taking paths as sequences of entities and relations, modeled with RNN



Part II: Models based on extra information

TOPICS:

- INCORPORATING ENTITY TYPES
- INCORPORATING TEXTUAL DESCRIPTIONS
- INCORPORATING RELATION PATHS
- <u>INCORPORATING LOGIC RULES</u>

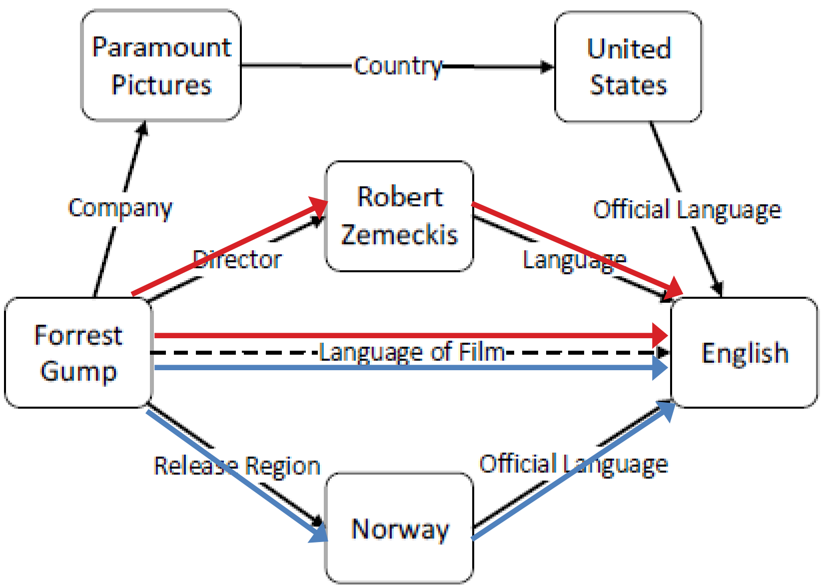
Logic rules

- First-order Horn clauses

$$\forall x, y: (x, \text{capitalOf}, y) \Rightarrow (x, \text{locatedIn}, y)$$

The capital city of a country must be located in that country

- Having a close relationship to relation paths



$$\forall x, y, z: (x, \text{Director}, y) \land (y, \text{Language}, z) \Rightarrow (x, \text{Language of Film}, z)$$

$$\forall x, y, z: (x, \text{Release Region}, y) \land (y, \text{Official Language}, z) \Rightarrow (x, \text{Language of Film}, z)$$

Example from: Lin et al. (2015). Modeling relation paths for representation learning of knowledge bases. EMNLP'15.

Hard rules vs. soft rules

Hard rules

- Rules that always hold with no exception

$$\forall x, y: (x, \text{capitalOf}, y) \Rightarrow (x, \text{locatedIn}, y)$$

*The capital city of a country **must** be located in that country*

- Usually requiring extensive manual effort to create or validate

Soft rules

- Rules with different confidence levels that can handle uncertainty

$$\forall x, y: (x, \text{bornIn}, y) \Rightarrow (x, \text{nationality}, y) \quad \text{(confidence = 0.8)}$$

*A person is **very likely (but not necessarily)** to have a nationality of the country where he/she was born*

- Automatically extracted via modern rule mining systems

Automatically extracted soft rules

- Soft rules mined from YAGO by AMIE+ (Galárraga et al. 2015)

Rule	Precision in the unknown region	Std. Confidence	PCA Confidence	New Predictions	Total predictions
?e <isMarriedTo> ?a ?e <hasChild> ?b => ?a <hasChild> ?b	34.48%	57.57%	57.57%	1865	1865
?a <isMarriedTo> ?f ?f <hasChild> ?b => ?a <hasChild> ?b	33.33%	56.11%	56.11%	1068	1986
?b <isMarriedTo> ?a => ?a <isMarriedTo> ?b	100.00%	53.01%	91.79%	5158	5158
?a <created> ?b ?a <produced> ?b => ?a <directed> ?b	3.33%	49.83%	58.68%	976	976
?a <actedIn> ?b ?a <created> ?b => ?a <directed> ?b	3.45%	38.29%	45.23%	518	608
?a <isMarriedTo> ?c ?c <livesIn> ?b => ?a <livesIn> ?b	75.00%	33.71%	69.88%	326	326
?a <dealsWith> ?f ?f <dealsWith> ?b => ?a <dealsWith> ?b	20.00%	28.19%	28.19%	75	75
?c <dealsWith> ?b ?a <dealsWith> ?c => ?a <dealsWith> ?b	26.67%	28.19%	28.19%	0	75
?e <isMarriedTo> ?a ?e <livesIn> ?b => ?a <livesIn> ?b	100.00%	22.81%	68.42%	299	542
?a <actedIn> ?b ?a <created> ?b => ?a <produced> ?b	18.52%	20.64%	38.98%	790	790
?a <directed> ?b => ?a <created> ?b	24.14%	18.75%	32.46%	26980	26980
?a <livesIn> ?f ?f <hasCapital> ?b => ?a <livesIn> ?b	16.67%	18.53%	18.53%	2851	2852
?a <livesIn> ?f ?f <isLocatedIn> ?b => ?a <livesIn> ?b	89.66%	17.80%	17.80%	4225	4227
?a <hasChild> ?c ?b <hasChild> ?c => ?a <isMarriedTo> ?b	76.67%	17.32%	40.98%	1170	1720
?a <created> ?b ?a <directed> ?b => ?a <produced> ?b	0.00%	16.11%	37.47%	4973	5258
?c <dealsWith> ?a ?c <dealsWith> ?b => ?a <dealsWith> ?b	42.86%	15.44%	21.78%	471	473
?c <hasOfficialLanguage> ?b ?c <isLocatedIn> ?a => ?a <hasOfficialLanguage> ?b	33.33%	14.29%	64.29%	100	100
?b <dealsWith> ?a => ?a <dealsWith> ?b	100.00%	13.64%	18.32%	152	219
?b <hasCapital> ?f ?a <livesIn> ?f => ?a <livesIn> ?b	48.28%	13.49%	13.49%	2856	3209
?c <hasAcademicAdvisor> ?a ?c <graduatedFrom> ?b => ?a <worksAt> ?b	6.67%	13.44%	39.04%	552	552
?a <imports> ?b => ?a <exports> ?b	3.33%	13.10%	14.47%	94	94
?a <produced> ?b => ?a <directed> ?b	0.00%	11.26%	13.40%	16262	17188
?a <dealsWith> ?c ?c <imports> ?b => ?a <imports> ?b	46.43%	10.58%	14.88%	438	438
?c <isCitizenOf> ?b ?a <influences> ?c => ?a <isCitizenOf> ?b	3.33%	10.56%	51.96%	1199	1199
?a <produced> ?b => ?a <created> ?b	10.34%	10.06%	15.69%	16272	17463
?c <dealsWith> ?a ?c <imports> ?b => ?a <imports> ?b	26.67%	9.28%	14.10%	271	355
?a <worksAt> ?b => ?a <graduatedFrom> ?b	3.57%	8.99%	16.25%	2217	2217
?a <diedIn> ?b => ?a <wasBornIn> ?b	0.00%	8.85%	22.53%	20196	20196
?a <exports> ?b => ?a <imports> ?b	3.45%	8.68%	10.33%	380	393

Results taken from: http://resources.mpi-inf.mpg.de/yago-naga/amie/data/yago2/amie_yago2.html

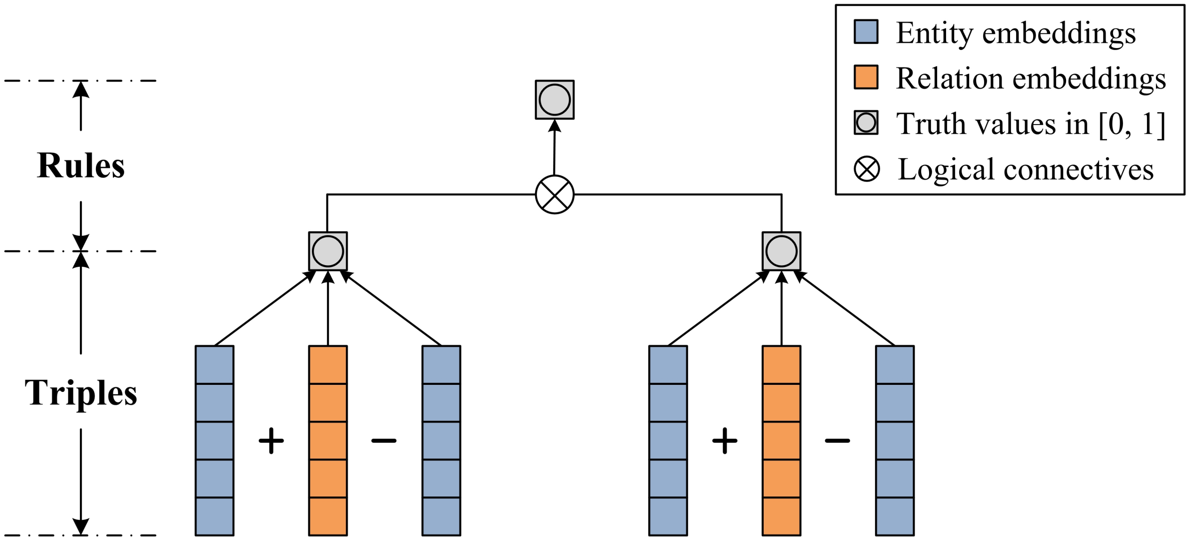
Jointly embedding with logic rules (Guo et al., 2016)

Key idea

- Jointly embedding subject-relation-object triples and ***hard*** rules

Jointly embedding framework

- Triples: Atomic formulae modeled by translation assumption
- Rules: Complex formulae modeled by t-norm fuzzy logics



(Paris, Capital-Of, France) ⇒ (Paris, Located-In, France)

Jointly embedding with logic rules (cont.)

- Modeling triples
  - Translation assumption: $\mathbf{s} + \mathbf{r} \approx \mathbf{o}$

$$I(s, r, o) = 1 - \frac{1}{3\sqrt{d}} \|\mathbf{s} + \mathbf{r} - \mathbf{o}\|_{\ell_1} \in [0, 1]$$

- Modeling rules
  - T-norm fuzzy logics: Truth value of a complex formulae is a composition of truth values of its constituents

$$I(f_1 \Rightarrow f_2) = I(f_1) \cdot I(f_2) - I(f_1) + 1$$

$$I(f_1 \wedge f_2 \Rightarrow f_3) = I(f_1) \cdot I(f_2) \cdot I(f_3) - I(f_1) \cdot I(f_2) + 1$$

- Joint learning
  - Minimizing a global loss over both triples and rules

$$\min \sum_{f^+ \in \mathcal{F}^+} \sum_{f^- \in \mathcal{F}^-} \max\left(0, \gamma - I(f^+) + I(f^-)\right)$$

f^+ and f^- can be either atomic or complex formulae

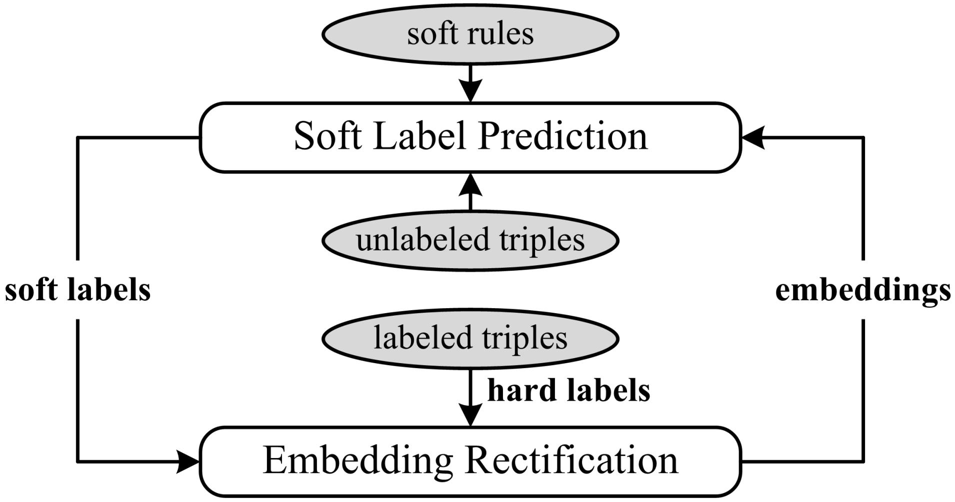
Rule-guided embedding (Guo et al., 2018)

Key idea

- Knowledge graph embedding with iterative guidance from ***soft*** rules

Iterative learning framework

- Soft label prediction: Use current embeddings and soft rules to predict soft labels for unlabeled triples
- Embedding rectification: Integrate both labeled and unlabeled triples to update current embeddings



Learning resources

- Labeled triples $\mathcal{L} = \{(x_\ell, y_\ell)\}$
- Unlabeled triples $\mathcal{U} = \{x_u\}$
- Soft rules $\mathcal{F} = \{(f_p, \lambda_p)\}$ and groundings $\mathcal{G} = \{g_{pq}\}$

Rule-guided embedding (cont.)

Soft label prediction

$$\min \frac{1}{2} \sum_{x_u \in \mathcal{U}} (s(x_u) - \phi(x_u))^2 + C \sum_{p,q} \xi_{pq}$$

$$\text{s.t. } \lambda_p (1 - \pi(g_{pq}|\mathcal{S})) \le \xi_{pq}, \ \forall g_{pq} \in \mathcal{G}$$

- $s(x_u)$: soft label to be predicted
- $\phi(x_u)$: truth value computed by current embeddings
- λ_p: confidence level of soft rule
- ξ_{pq}: slackness to handle uncertainty

① soft label should stay close to truth value

② groundings of rule f_p should hold with confidence λ_p

Embedding rectification

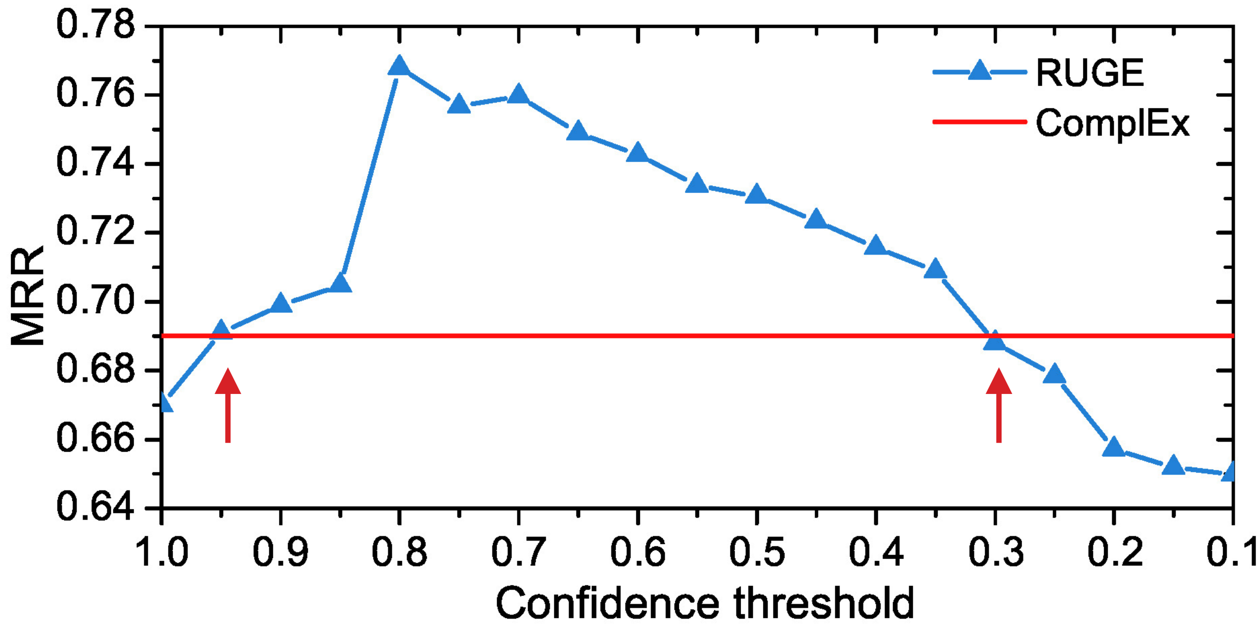
$$\min \frac{1}{|\mathcal{L}|} \sum_{\mathcal{L}} \ell(\phi(x_\ell), y_\ell) + \frac{1}{|\mathcal{U}|} \sum_{\mathcal{U}} \ell(\phi(x_u), s(x_u))$$

② loss of labeled triples with their hard labels

② loss of unlabeled triples with their soft labels

Rule-guided embedding (cont.)

- Influence of confidence levels of soft rules on link prediction



Soft rules (even those with moderate confidence levels) are highly beneficial

Main obstacle: Propositionalization

- First-order rules have to be propositionalized using entities in knowledge graphs (grounding)
  - First-order rule

$$\forall x, y: (x, \text{capitalOf}, y) \Rightarrow (x, \text{locatedIn}, y)$$

⬇ grounding

  - Proposition rules

$$(\text{Paris}, \text{capitalOf}, \text{France}) \Rightarrow (\text{Paris}, \text{locatedIn}, \text{France})$$

$$(\text{Rome}, \text{capitalOf}, \text{Italy}) \Rightarrow (\text{Rome}, \text{locatedIn}, \text{Italy})$$

$$(\text{Berlin}, \text{capitalOf}, \text{Germany}) \Rightarrow (\text{Berlin}, \text{locatedIn}, \text{Germany})$$

$$(\text{Beijing}, \text{capitalOf}, \text{China}) \Rightarrow (\text{Beijing}, \text{locatedIn}, \text{China})$$

$$(\text{Moscow}, \text{capitalOf}, \text{Russia}) \Rightarrow (\text{Moscow}, \text{locatedIn}, \text{Russia})$$

$$\vdots$$

Scales exponentially with graph size (number of entities)

To avoid grounding: Regularizing relation embeddings

- Key idea
  - Modeling first-order rules by regularizing relation embeddings (using no entity embeddings)

First-order rule

$$\forall x, y:\ (x, r_p, y) \Rightarrow (x, r_q, y)$$

Any two entities linked by relation r_p should also be linked by r_q

Equivalent statement

$$\forall s, o \in \mathcal{E}:\ f(s, r_p, o) \leq f(s, r_q, o)$$

For any two entities s and o, if (s, r_p, o) is a valid triple with a high score, then (s, r_q, o) with an even higher score will also be predicted as valid by the embedding model

To avoid grounding: Regularizing relation embeddings (cont.)

- Applying to entity pair model (Demeester et al., 2016)
  - Triple scoring function: $f(s,r,o) = \mathbf{v}_{so}^{\mathrm{T}}\mathbf{v}_r$
  - **Non-negative** entity pair representation: $\mathbf{v}_{so} \geq 0, \forall s,o \in \mathcal{E}$

$$\boxed{\mathbf{v}_{r_p} \leq \mathbf{v}_{r_q}} \quad \Rightarrow \quad \forall s,o \in \mathcal{E}\colon f(s,r_p,o) \leq f(s,r_q,o)$$

- Applying to ComplEx (Ding et al., 2018)
  - Triple scoring function: $f(s,r,o) = \mathrm{Re}\left(\mathbf{s}^{\mathrm{T}}\mathrm{diag}(\mathbf{r})\bar{\mathbf{o}}\right)$
  - **Non-negative** entity representation: $\mathrm{Re}(\mathbf{e}) \geq 0, \mathrm{Im}(\mathbf{e}) \geq 0,\ \forall e \in \mathcal{E}$

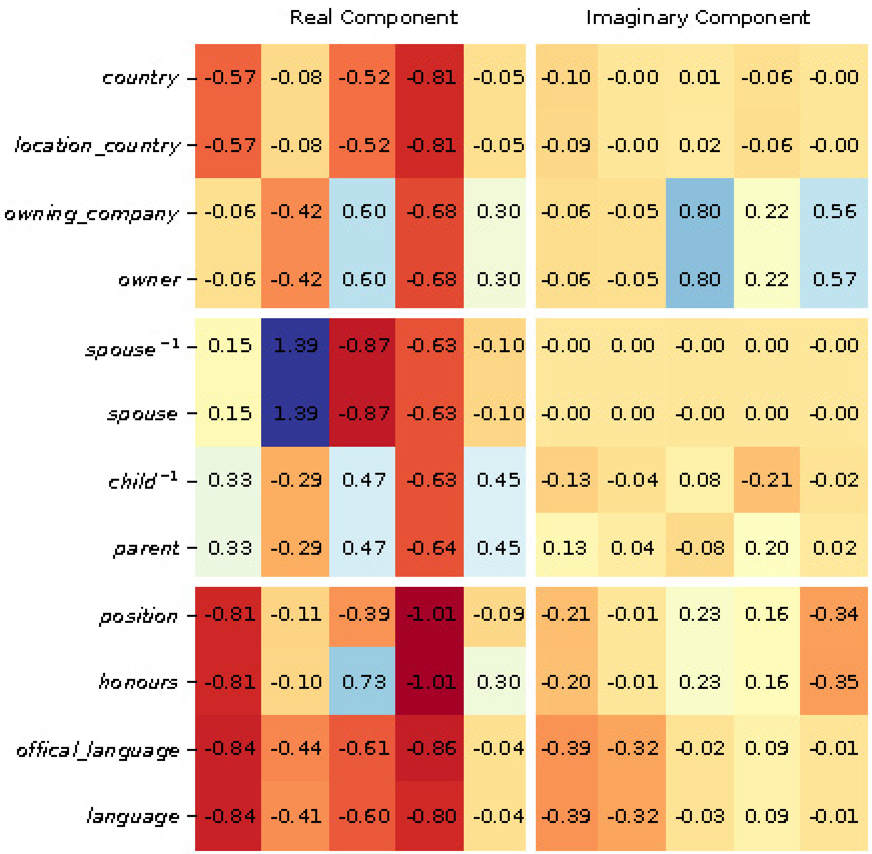
$$\boxed{\begin{array}{c}\mathrm{Re}(\mathbf{r}_p) = \mathrm{Re}(\mathbf{r}_q) \\ \mathrm{Im}(\mathbf{r}_p) \leq \mathrm{Im}(\mathbf{r}_q)\end{array}} \quad \Rightarrow \quad \forall s,o \in \mathcal{E}\colon f(s,r_p,o) \leq f(s,r_q,o)$$

Pros: Complexity independent of graph size

Cons: Can only handle rules in the simplest form $\forall x,y\colon (x,r_p,y) \Rightarrow (x,r_q,y)$

To avoid grounding: Regularizing relation embeddings (cont.)

- Visualization of relation embeddings learned by regularizing ComplEx (Ding et al., 2018)



Equivalence

$$\mathrm{Re}(\mathbf{r}_p) = \mathrm{Re}(\mathbf{r}_q)$$
$$\mathrm{Im}(\mathbf{r}_p) = \mathrm{Im}(\mathbf{r}_q)$$

Inversion

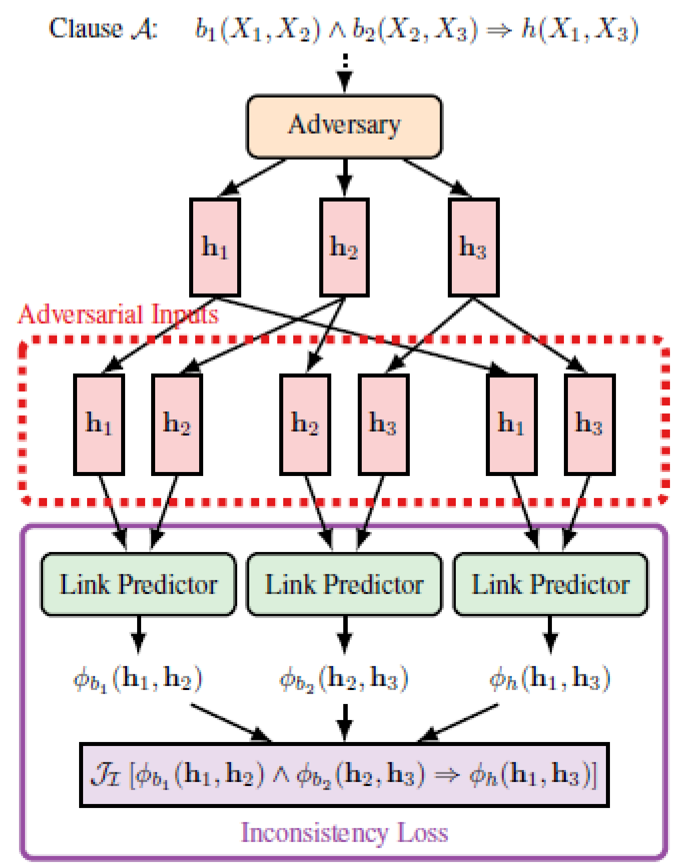
$$\mathrm{Re}(\mathbf{r}_p) = \mathrm{Re}(\mathbf{r}_q)$$
$$\mathrm{Im}(\mathbf{r}_p) = -\mathrm{Im}(\mathbf{r}_q)$$

Implication

$$\mathrm{Re}(\mathbf{r}_p) \leq \mathrm{Re}(\mathbf{r}_q)$$
$$\mathrm{Im}(\mathbf{r}_p) = \mathrm{Im}(\mathbf{r}_q)$$

To avoid grounding: Adversarial training (Minervini et al., 2017)

- Key idea
  - Modeling first-order rules with **adversarially generated** entities rather than real entities
- Adversarial training architecture
  - Adversary: Generate a set of adversarial entity embeddings on which the rules are violated most
  - Discriminator: Learn an embedding model compatible with real input (triples) while satisfying the rules on the adversarial set



Generating rather than traversing entities, with complexity independent of graph size

Review

Problem

- To learn distributed representations of entities and relations from extra information beyond subject-relation-object triples

Incorporating entity types

- Difficulty: hierarchical types and multiple type labels

Incorporating textual descriptions

- Jointly embedding knowledge graphs and words
- Modeling entity embedding as a composition of word embeddings

Incorporating relation paths

- Taking path as a sequence of relations, sequence modeling by addition, multiplication, or RNN
- Intermediate entities may also be included during sequence modeling

Review (cont.)

Incorporating logic rules

- Hard rules versus soft rules
- Difficulty: propositionalization (grounding)
- Avoiding grounding by regularizing entity representations
- Avoiding grounding by adversarial training

References

[Das et al., 2017] Das, R., Neelakantan, A., Belanger, D., and McCallum, A. (2017). Chains of reasoning over entities, relations, and text using recurrent neural networks. In EACL'17.

[Demeester et al., 2016] Demeester, T., Rocktäschel, T., and Riedel, S. (2016). Lifted rule injection for relation embeddings. In EMNLP'16.

[Ding et al., 2018] Ding, B., Wang, Q., Wang, B., and Guo, L. (2018). Improving knowledge graph embedding using simple constraints. In ACL'18.

[Galárraga et al., 2015] Galárraga, L. A., Teflioudi, C., Hose, K., and Suchanek, F. (2015). Fast rule mining in ontological knowledge bases with AMIE+. VLDB.

[Guo et al., 2015] Guo, S., Wang, Q., Wang, B., Wang, L., and Guo, L. (2015). Semantically smooth knowledge graph embedding. In ACL'15.

[Guo et al., 2016] Guo, S., Wang, Q., Wang, L., Wang, B., and Guo, L. (2016). Jointly embedding knowledge graphs and logical rules. In EMNLP'16.

[Guo et al., 2018] Guo, S., Wang, Q., Wang, L., Wang, B., and Guo, L. (2018). Knowledge graph embedding with iterative guidance from soft rules. In AAAI'18.

[Guu et al., 2015] Guu, K., Miller, J., and Liang, P. (2015). Traversing knowledge graphs in vector space. In EMNLP'15.

[Lin et al., 2015] Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., and Liu, S. (2015). Modeling relation paths for representation learning of knowledge bases. In EMNLP'15.

References

[Minervini et al., 2017] Minervini, P., Demeester, T., Rocktäschel, T., and Riedel, S. (2017). Adversarial sets for regularising neural link predictors. In UAI'17.

[Neelakantan et al., 2015] Neelakantan, A., Roth, B., and McCallum, A. (2015). Compositional vector space models for knowledge base completion. In ACL'15.

[Socher et al., 2013] Socher, R., Chen, D., Manning, C., and Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In NIPS'13.

[Wang et al., 2014] Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph and text jointly embedding. In EMNLP'14.

[Xie et al., 2016a] Xie, R., Liu, Z., Jia, J., Luan, H., and Sun, M. (2016a). Representation learning of knowledge graphs with entity descriptions. In AAAI'16.

[Xie et al., 2016b] Xie, R., Liu, Z., and Sun, M. (2016b). Representation learning of knowledge graphs with hierarchical types. In IJCAI'16.

[Zhong et al., 2015] Zhong, H., Zhang, J., Wang, Z., Wan, H., and Chen, Z. (2015). Aligning knowledge and text embeddings by entity descriptions. In EMNLP'15.