组会

曾双

2020.12.26

COLING 2020

- Improving Relation Extraction with Relational Paraphrase Sentences
- Logic-guided Semantic Representation Learning for Zero-Shot Relation

COLING 2020

Improving Relation Extraction with Relational Paraphrase Sentences

Junjie Yu[1], Tong Zhu[1], Wenliang Chen[1], Wei Zhang[2], Min Zhang[1]
Institute of Artificial Intelligence, School of Computer Science and Technology,
Soochow University, China[1]
Alibaba Group, China[2]
{jjyu,tzhu7}@stu.suda.edu.cn, {wlchen,minzhang}@suda.edu.cn,
lantu.zw@alibaba-inc.com

Motivation

- RE数据集规模小，不足以覆盖现实场景中各种各样的关系表达

(1) “Steve Jobs co-founded Apple Computer.”

(2) “Steve Jobs was the co-founder of Apple Computer.”

(3) “Steve Jobs started Apple Computer with Wozniak.”

- 解决办法
  - 人工标注更多数据——标签可靠但成本（时间、人力、金钱）高
  - 远程监督——规模可以很大，但有严重的标签错误问题

Motivation

- 提出通过relational paraphrase的方式来提供多样化关系表达
    - 多样性：多个Back-Translation系统生成句子的多个复述
    - 源句子和复述句的实体对齐：上下文相似度的词对齐方式
- 提出模型对多样化关系表达进行建模

Google Translation

Baidu Translation

Xiaoniu Translation

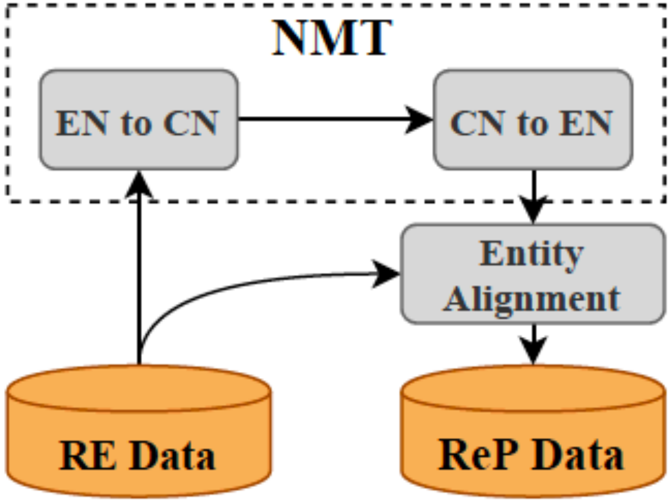


Figure 1: Framework of building the relational paraphrase data. EN=English, CN=Chinese.

Triple: <All Basotho Convention, org:founded_by, Tom Thabane >
#1 [tom thabane] , who set up the [all basotho convention] four months ago ...
#2 [tom *taba*] , who four months ago , formed a [*wholly* basotho] , ... #3 four months ago , [tom thabane] set up the [all *basoto conference*] , ... #4 [tom thabane] , who founded the [all *basoto congress*] four months ago , ...

Figure 2: An example from our ReP data. #1 is a human-annotated sentence, and #2-4 are paraphrase sentences. Blue words with underlines mean different clues for relation "org:founded_by" between two entities.

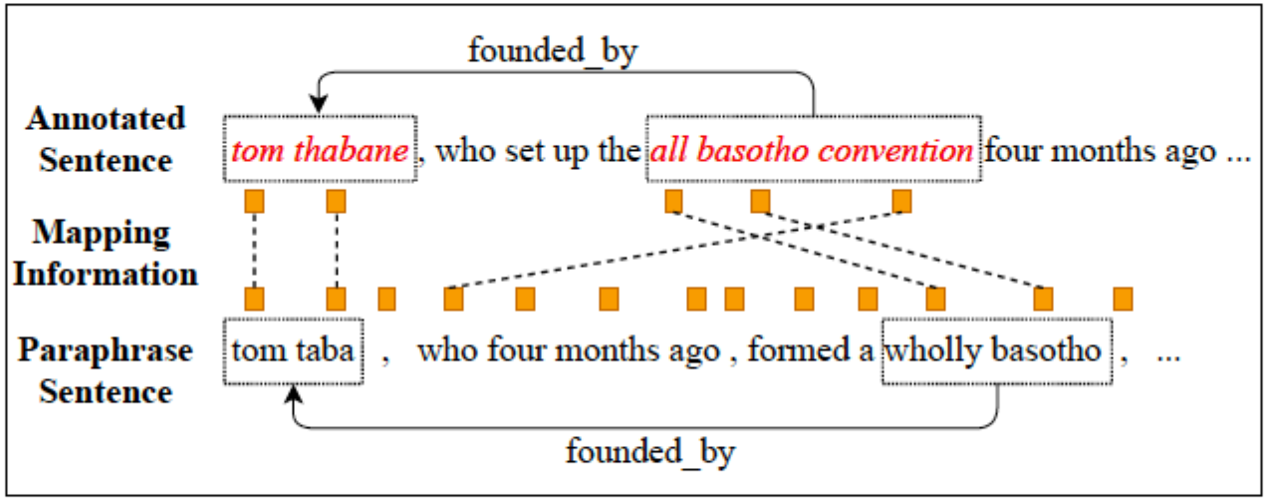


Figure 3: An example of aligning entities and relations.

Para.	**Acc.**	78.0%	
Entity	**Diff.**	**Yes**	**No**
	Prop.	47.4%	52.6%
	Acc.	89.2%	100.0%
Both	**Acc.**	74.0%	

Table 2: Manual evaluation of the ReP-AUTO. **Para.**: correct paraphrase. **Acc.**: accuracy. **Entity**: performance of entity alignment. **Prop.**: proportion. **Diff.**: whether entities have been changed.

$$s^{t_i} = \underset{s_j \in s}{\operatorname{argmax}} \{\cos(\mathbf{h}_i^t, \mathbf{h}_j^s)\}.$$

41 + 1种关系

Data Split	Train				Dev	Test
	# Sen	# Sen1	# Fact	# Sen1/Fact	# Sen	# Sen
Gold-Annotated (TACRED)	68,124	13,012	8,190	1.6	22,631	15,509
Auto-Generated	204,372	39,036	8,190	4.8	-	-

Table 1: Statistics of the RE data used in the experiments. # Sen: number of all sentences. # Sen1: number of sentences excluding sentences labeled with `no_relation`. # Fact: number of relation facts (excluding `no_relation`). # Sen1/Fact: average number of supporting sentences for each relation fact.

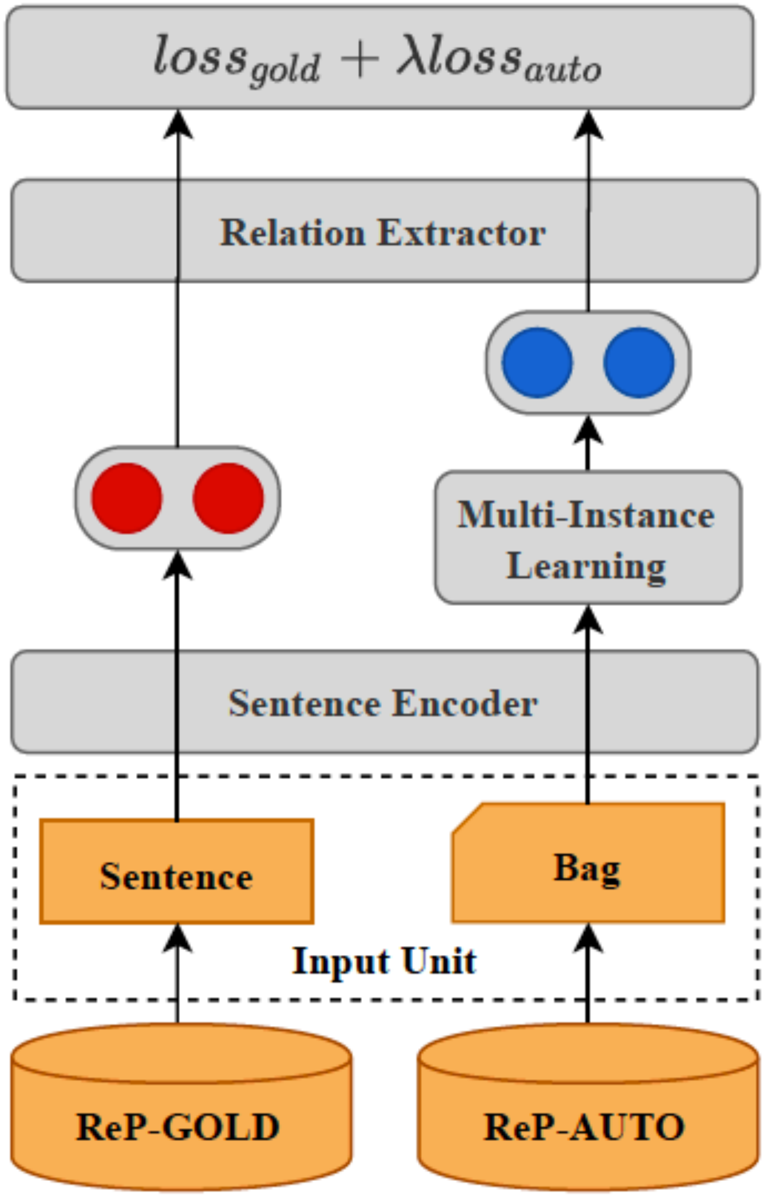


Figure 4: Training Framework

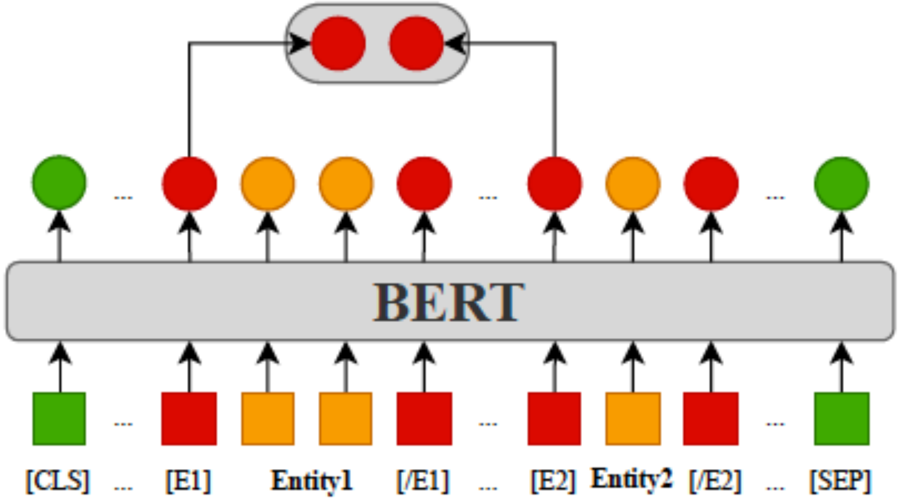


Figure 5: Sentence Encoder

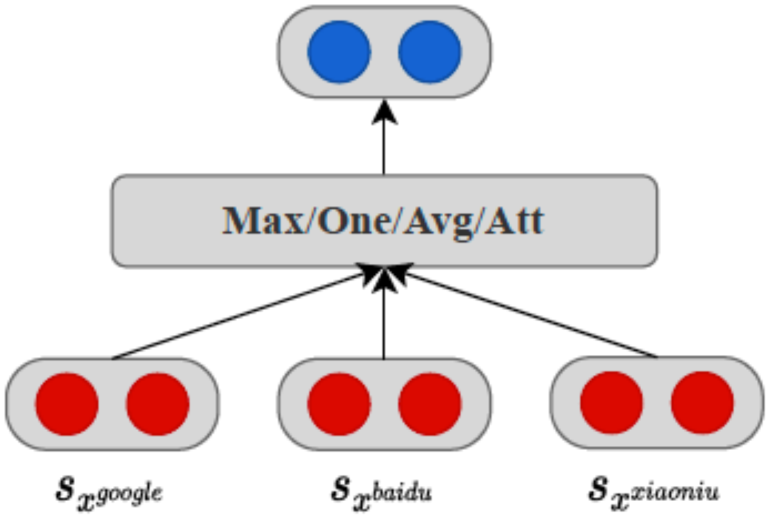


Figure 6: Multi-Instance Learning

Bag-Max. In this method, we generate the bag-level representations by performing maximum pooling on outputs of sentences in bag B:

$$\mathbf{s}_B = \underset{x \in B}{maxpool}(\mathbf{s}_x). \tag{5}$$

Bag-One. Different from outputting a maximum value on each dimension in Bag-Max method, Bag-One outputs the best representation from one of three sentences in B by calculating the probability on its gold relation type after a softmax layer.

$$\mathbf{s}_B = \mathbf{s}_{x'}, \tag{6}$$

$$x^{'} = \underset{x \in B}{\operatorname{argmax}}\, p(r_x | x, \theta),$$

where $p()$ outputs the probability of relation type r_x for the input sentence x under current model parameters θ.

Bag-Avg. Similar to Bag-Max, Bag-Avg method adds an averaged pooling layer after encoding sentences in B:

$$\mathbf{s}_B = \frac{1}{|B|} \sum_{x \in B} \mathbf{s}_x. \tag{7}$$

Bag-Att. Inspired by the attention mechanism used in Lin et al. (2016), we add an attention layer to output bag-level representations for sentences in B. First we generate attention weights α for sentences in B by calculating how well it matches with their gold relation type. Then, we output a weighted sum of representations:

$$\mathbf{s}_B = \sum_{x \in B} \alpha_x \mathbf{s}_x, \tag{8}$$

$$\alpha_x = \frac{exp(e_x)}{\sum_{x' \in B} exp(e_{x'})},$$
$$e_x = \mathbf{s}_x \mathbf{A} \mathbf{r},$$

where e_x measures how well $\mathbf{s}_x$ matches with the query vector $\mathbf{r} \in \mathbb{R}^{2d}$ which is the representation of the gold relation of x, and $\mathbf{A} \in \mathbb{R}^{2d \times 2d}$ represents a diagonal matrix.

Systems	F1
Baseline (ReP-GOLD)	68.67
ReP-AUTO	66.75
ReP-GOLD ∪ ReP-AUTO	68.53
ReP-GOLD + Google	69.37
ReP-GOLD + Baidu	69.12
ReP-GOLD + Xiaoniu	69.24
ReP-GOLD + Bag-Max	69.45
ReP-GOLD + Bag-One	69.46
ReP-GOLD + Bag-Avg	**69.60**
ReP-GOLD + Bag-Att	69.38

Table 4: Comparison with Baseline on test set.

Systems	F1
CNN-PE † (Zeng et al., 2014)	61.1
PCNN † (Zeng et al., 2015)	62.0
SDP-LSTM ‡ (Xu et al., 2015)	58.7
Tree-LSTM ‡ (Tai et al., 2015)	62.7
PA-LSTM (Zhang et al., 2017)	65.1
SA-LSTM+D (Yu et al., 2019)	67.6
C-GCN + PA-LSTM (Zhang et al., 2018)	68.2
MTB on $BERT_{large}$ (Soares et al., 2019)	**71.5**
Baseline on $BERT_{base}$	68.7
ReP-GOLD + Bag-Avg on $BERT_{base}$	69.6
Baseline on $BERT_{large}$	70.2
ReP-GOLD + Bag-Avg on $BERT_{large}$	**70.8**

Table 5: Comparion with previous results. † marks results reported in Yu et al. (2019); ‡ marks results reported in Zhang et al. (2017).

COLING 2020

Logic-guided Semantic Representation Learning for Zero-Shot Relation Classification

Juan Li[1,2*], **Ruoxu Wang**[1,2*], **Ningyu Zhang**[1,2*†], **Wen Zhang**[1,2], **Fan Yang**[1,2], **Huajun Chen**[1,2 †]
[1] Zhejiang University
[2] AZFT Joint Lab for Knowledge Engine
{lijuan18,ruoxuwang,zhangningyu,wenzhang2015,21821249,huajunsir}@zju.edu.cn

Zero-shot RC

- 从句子中预测实体间没有见过的关系类别（零样本）
  - 更符合现实场景（存在大量细粒度的关系类别）
- 前人方法
  - 阅读理解(Levy et al., 2017)：构造针对关系类别的问题达到zero-shot的效果
  - 文本蕴含(Obamuyide and Vlachos, 2018)：句子为前提，关系三元组的描述为假设
  - 依赖人力构造问题和关系的描述

Zero-shot RC

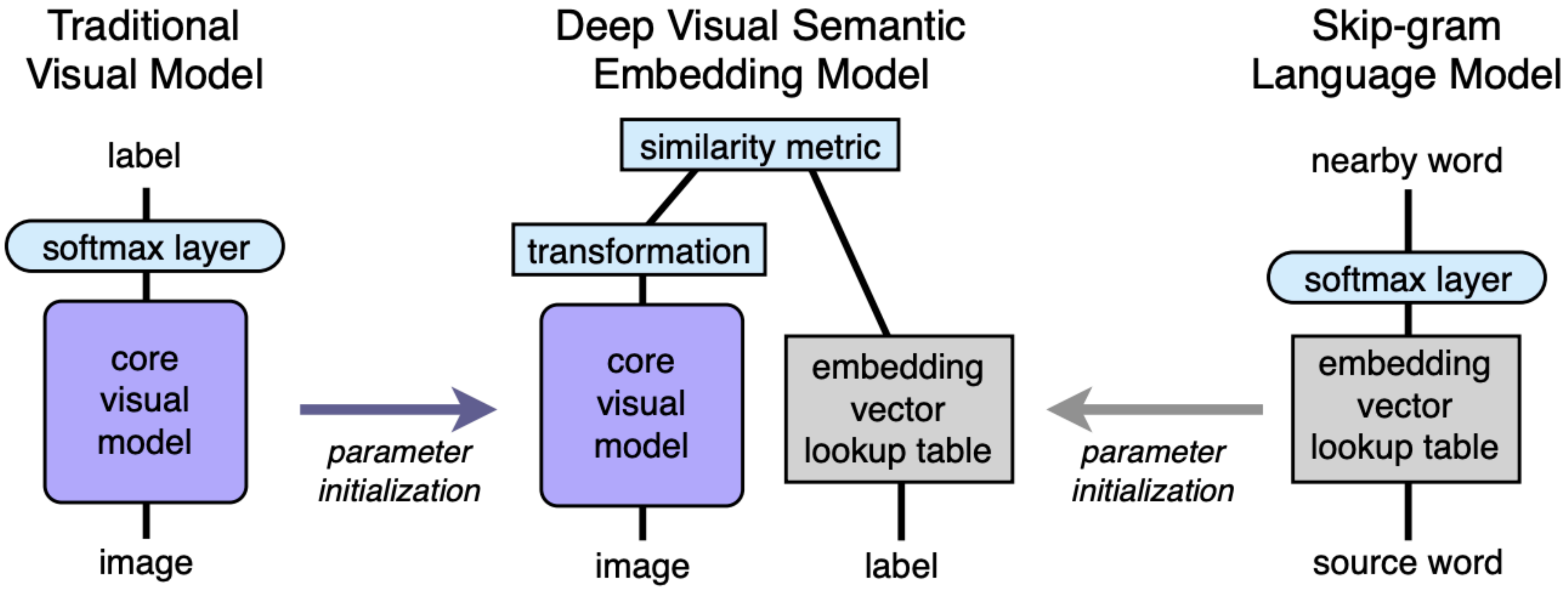
Relation	Question Template
$educated\_at(x, y)$	Where did x graduate from? In which university did x study? What is x's alma mater?
$occupation(x, y)$	What did x do for a living? What is x's job? What is the profession of x?
$spouse(x, y)$	Who is x's spouse? Who did x marry? Who is x married to?

Figure 1: Common knowledge-base relations defined by natural-language question templates.

Relation	Subject (**X**)	Object (**Y**)	Text (Premise)	Description (Hypothesis)
religious_order	Lorenzo Ricci	Society of Jesus	**X** (August 1, 1703 – November 24, 1775) was an Italian Jesuit, elected the 18th Superior General of the **Y**.	*X was a member of the group Y*
director	Kispus	Erik Balling	**X** is a 1956 Danish romantic comedy written and directed by **Y**.	*The director of X is Y*
designer	Red Baron II	Dynamix	**X** is a computer game for the PC, developed by **Y** and published by Sierra Entertainment.	*Y is the designer of X*

Zero-shot Learning

- CV
  - 学习输入样本的特征空间到标签语义空间的映射 （DeViSE， ConSE）



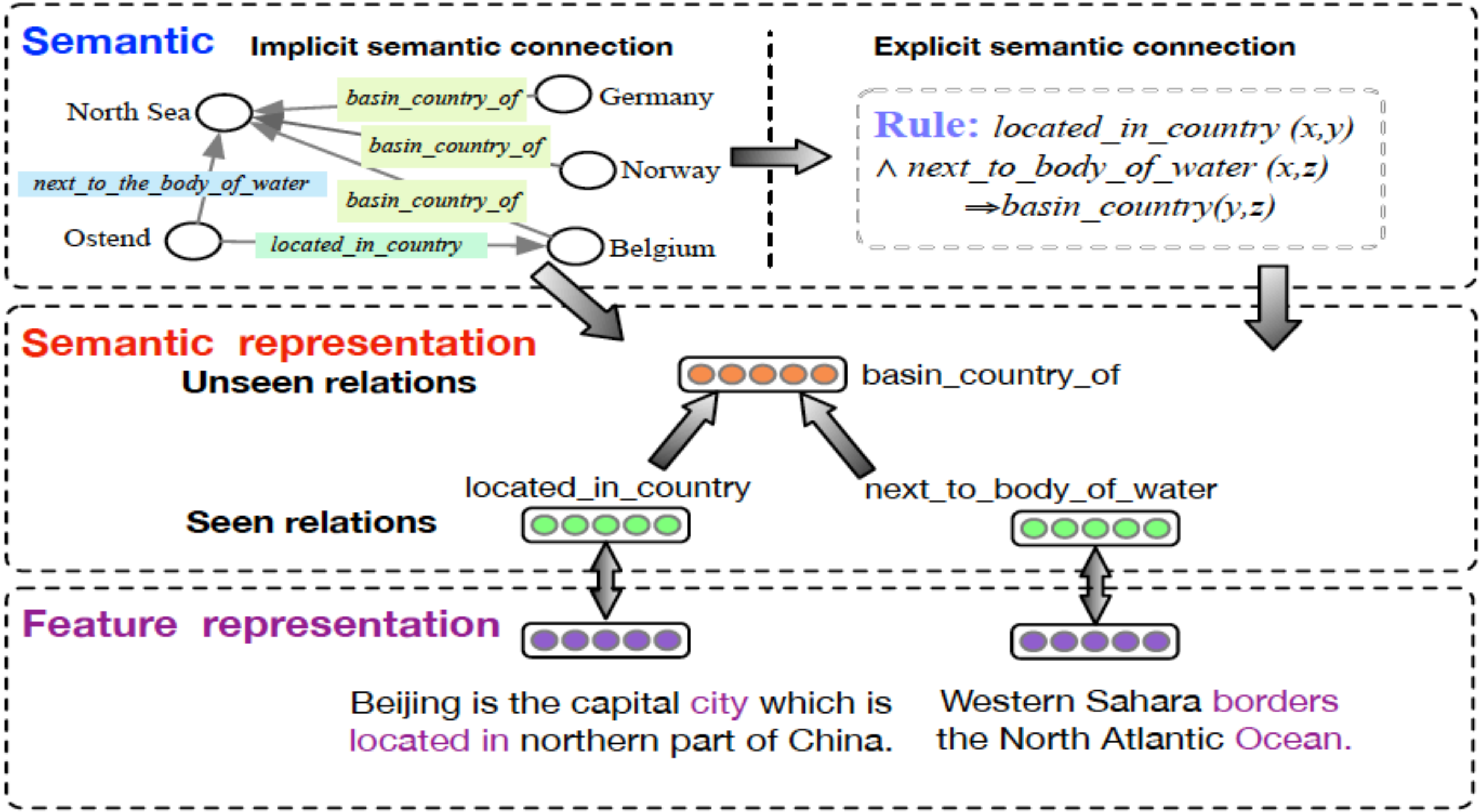
动机

- 前人使用word embedding作为标签的语义空间
- 但忽略了关系之间的语义联系
  - 隐式：Knowledge Graph Embedding (KGE)，相似关系在空间中更接近
  - 显式：Rule Learning (RL)，人类使用符号推理，基于已知关系识别未知关系
    - If ***located in country (x, y)*** and ***next to body of water (x,z)***

    Then ***basin country of (y,z)***

动机

- 前人使用word embedding作为标签的语义空间
- 但忽略了关系之间的语义联系
  - 隐式：Knowledge Graph Embedding (KGE)，相似关系在空间中更接近
    - 如TranE，只依赖于KG的结构信息得到关系表示，不使用word信息
  - 显式：Rule Learning (RL)，人类使用符号推理，基于已知关系识别未知关系
    - 从大规模KG中抽取规则



Semantic
Implicit semantic connection
Explicit semantic connection
North Sea
Germany
Norway
Belgium
Ostend
basin_country_of
basin_country_of
basin_country_of
next_to_the_body_of_water
located_in_country
Rule: located_in_country (x,y)
∧ next_to_body_of_water (x,z)
⇒basin_country(y,z)
Semantic representation
Unseen relations
basin_country_of
located_in_country
next_to_body_of_water
Seen relations
Feature representation
Beijing is the capital city which is located in northern part of China.
Western Sahara borders the North Atlantic Ocean.

方法

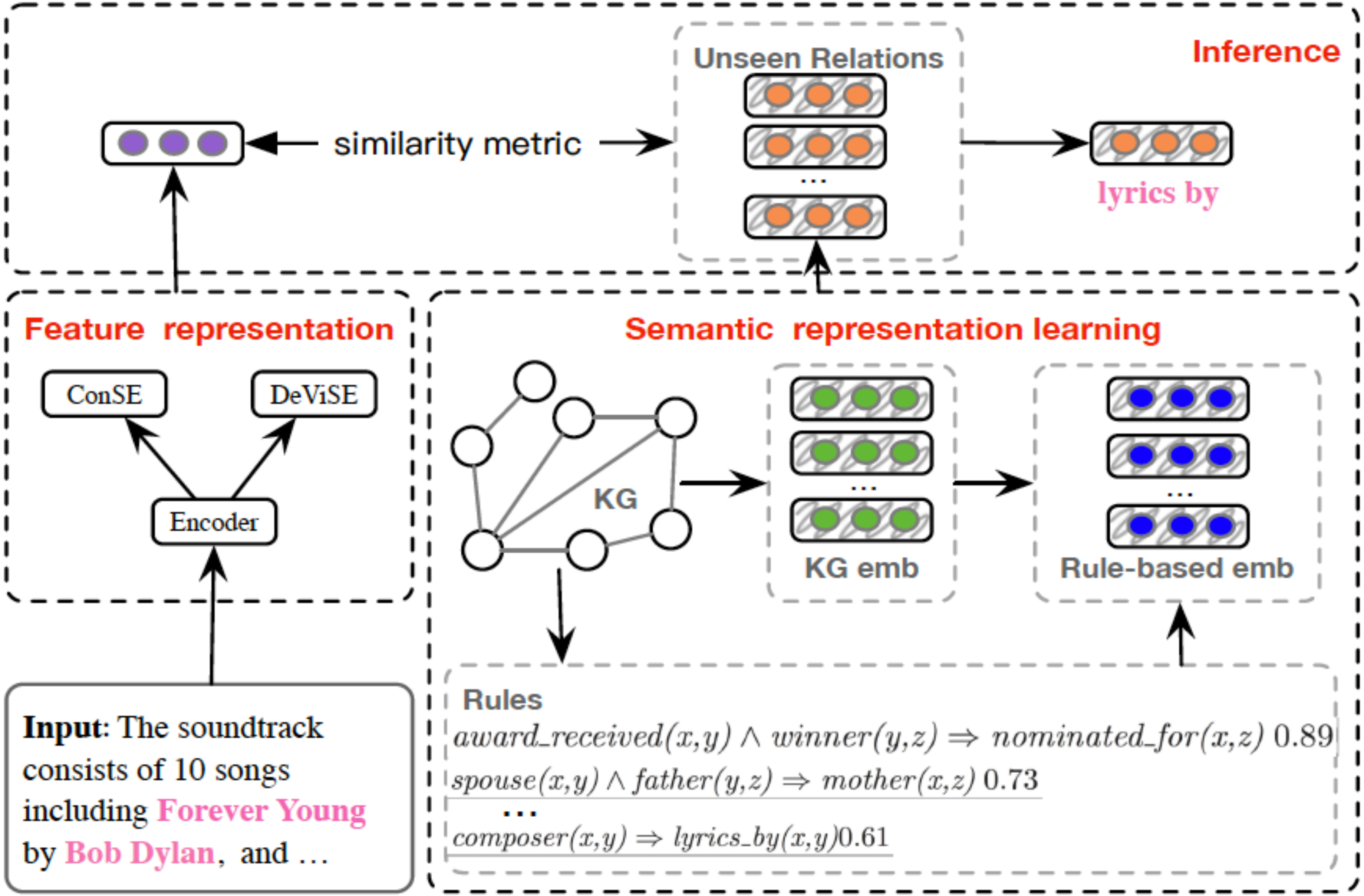


Figure 2: The architecture of Logic-guided Semantic Representation Learning model.

Feature Representation

$$f = PCNN(x_1, ..., x_n)$$

DeViSE

$$g = W * f + b$$

$$R_t^S, p_t, E(R_t^S) = C(f), t = 1, ..., T$$

ConSE

$$g = \sum_{t=1}^{T} p_t * E(R_t^S)$$

KG Embedding

$$E_{kw} = W_2 * ([E_{kg}; E_{wd}] + b_2)$$

Rule-guided Embedding

- 使用经典的规则挖掘方法AMIE，从KG中生成规则以及他们对应的PCA置信度

$$E_{rl}(R_i^U) = \frac{\sum_{j=1}^{K} conf_j * E_{kg}(Rule_{ij}^U)}{\sum_{j=1}^{K} conf_j}$$

with two rules about unseen relation r, $R1 : r_A \wedge r_B \Rightarrow r$ and $R2 : r_C \wedge r \Rightarrow r_D$, following TransE's assumption, we calculate embedding of r via $E_{rl}(r) = \frac{conf_1 * [E_{kg}(r_A) + E_{kg}(r_B)] + conf_2 * [E_{kg}(r_D) - E_{kg}(r_C)]}{conf_1 + conf_2}$.

Inference

$$\overline{y_i} = sim(f_{x_i}, E(R_{x_i}^U))$$

Results

	ConSE(Hit@n)			DeViSE(Hit@n)		
	1	2	5	1	2	5
$+E_{wd}$	0.21	0.30	0.43	0.11	0.19	0.39
$+E_{kg}$	0.39	0.53	0.69	0.22	0.38	0.57
$+E_{rl}$	0.40	0.54	0.72	0.23	0.39	0.58
$+E_{kw}$	0.39	0.55	0.72	0.23	**0.40**	**0.59**
$+E_{rw}$	0.40	0.55	0.70	0.23	0.34	0.57
$+E_{kr}$	**0.43**	**0.57**	**0.74**	**0.25**	0.39	**0.59**

Results

Unseen Relations	F1-score		Top 3 Related Seen Relations	
	$+E_{kg}$	$+E_{wd}$	$+E_{kg}$	$+E_{wd}$
lyrics_by	**0.52**	0.06	performer	influenced_by
			composer	spouse
			cast_member	cast_member
after_a_work_by	**0.51**	0.01	author	named_after
			screenwriter	author
			creator	characters
location_of_formation	**0.46**	0.02	headquarters_location	subclass_of
			location	opposite_of
			capital	part_of
nominated_for	**0.97**	0.56	award_received	award_received
			winner	part_of
			participant_of	member_of
mother	0.40	**0.83**	follows	child
			spouse	spouse
			twinned_administrative_body	father
developer	0.38	**0.49**	publisher	manufacturer
			manufacturer	publisher
			owned_by	owned_by
office_contested	**0.26**	0.00	position_held	————
			successful_candidate	
			applies_to_jurisdiction	
occupant	**0.31**	0.00	owned_by	————
			location	
			headquarters_location	
drafted_by	**0.81**	0.00	member_of_sports_team	————
			educated_at	
			member_of	

Results

Unseen Relations	F1-score						Related rules w.r.t. unseen relations
	$+E_{wd}$	$+E_{kg}$	$+E_{rl}$	$+E_{kw}$	$+E_{rw}$	$+E_{kr}$	
mother	**0.83**	0.40	0.77	0.53	0.80	0.78	*mother(x,z) ⇐ spouse(x,y) ∧ father(y,z)* *mother(x,y) ⇐ child(y,x)*
lyrics_by	0.06	0.52	0.51	0.49	0.48	**0.52**	*lyrics_by(x,y) ⇐ composer(x,y)*
nominated_for	0.56	**0.97**	0.96	**0.97**	0.96	0.96	*nominated_for(x,z) ⇐ award_received(x,y) ∧ winner(y,z)*
producer	0.41	0.52	**0.55**	0.54	0.52	0.53	*producer(x,y) ⇐ director(x,y)* *producer(x,y) ⇐ screenwriter(x,y)* *producer(x,y) ⇐ cast_member(x,y)*
field_of_work	0.04	0.14	0.29	0.11	0.29	**0.37**	*field_of_work(x,y) ⇐ occupation(x,y)*
connecting_line	0.00	0.10	0.43	0.28	0.42	**0.47**	*connecting_line(x,z) ⇐ adjacent_station(y,x) ∧ part of(y,z)*
residence	0.01	0.32	0.30	0.30	0.38	**0.39**	*residence(x,y) ⇐ place_of_birth(x,y)* *residence(x,y) ⇐ place_of_death(x,y)*

Table 4: Results of all different embeddings on F1 score when regrading ConSE as project funtion, and related rules w.r.t unseen relations.

Thanks!