Joint Extraction of Entities and Relations Based on a Novel Graph Scheme

潘佳鑫

Abstract

- Most existing neural joint methods extract entities and relations separately and achieve joint learning through parameter sharing.
- Directed graph
- Transition-based approach
- Achieve joint learning through joint decoding
- Dependencies not only between entities and relations, but also between relations.

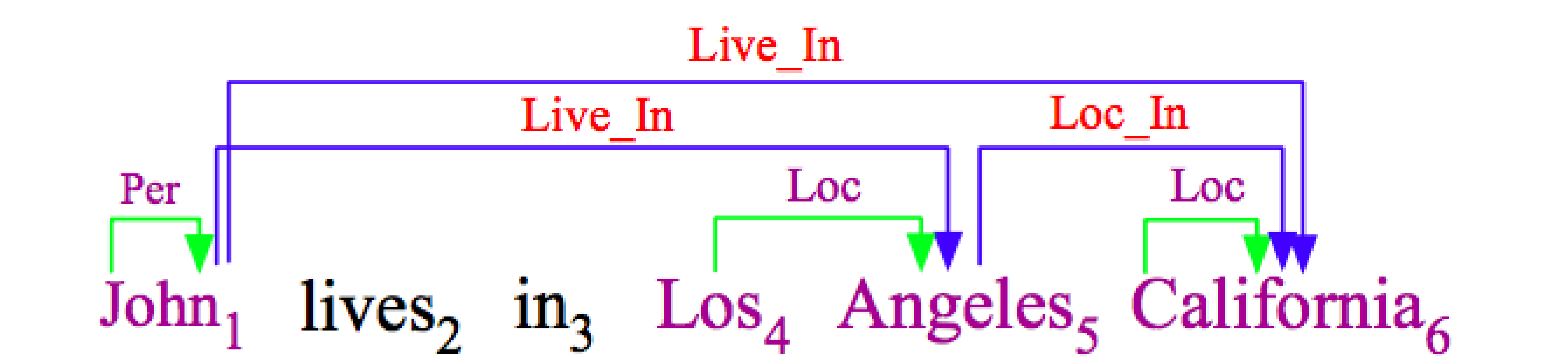
Introduction

- Pipeline : entity recognition and relation extraction , neglects the relevance between these two sub-tasks
- Joint model : statistical methods , neural methods
- Statistical methods: heavily rely on complicated feature engineering
- Neural methods: parameter sharing , no explicit features are used to model output-output dependencies

Introduction

- Zheng *et al.* [2017] designs a novel tagging scheme
- Indirectly captures output structural correspondences, and is incapable of identifying overlapping relations (e.g. one entity can only have at most one relation).
- A novel graph scheme, nodes may have multiple or no heads
- A novel transition system to generate the directed graph.
- A special recursive neural network to better model underlying *entity-relation* and *relation-relation* dependencies.

Introduction



The Graph Scheme

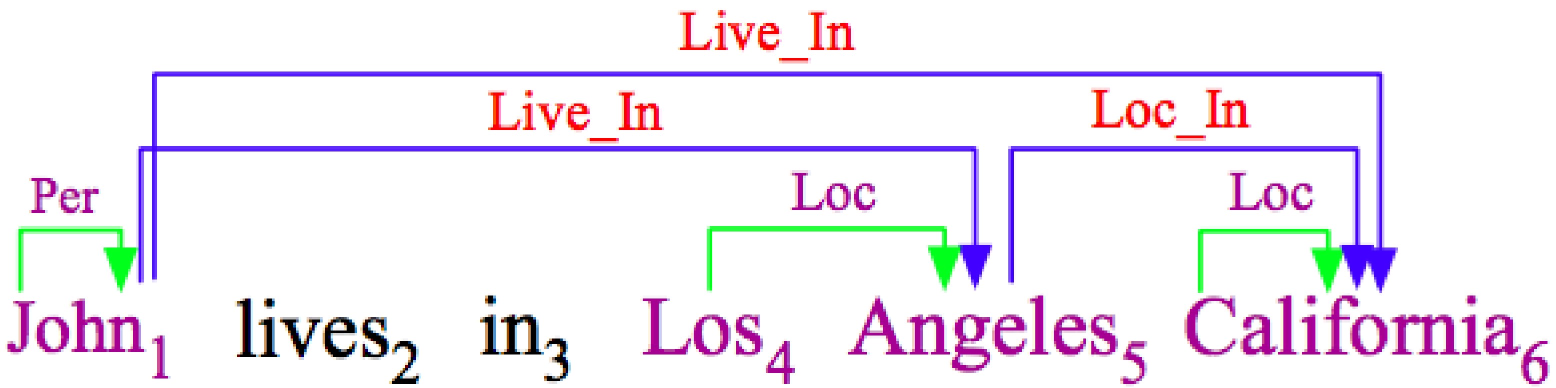
- The nodes in the graph correspond to words in the input sentence
- The directed arcs are broadly categorized into: 1) entity arcs that represent internal structures of entities; 2) relation arcs that represent relations between entities, where head node means the first element of relation and modifier node means the second element of relation.
- The other words irrelevant to the final result have no corresponding arcs.

Transition System

- 1) entity actions, which are used to recognize entities; 2) relation actions, which are used to recognize relations between entities.
- Formally, we use a tuple (σ, δ, e, β, R, E) to represent each state, where σ is a stack holding processed entities, δ is a stack holding entities that are popped out of σ but will be pushed back in the future, e is a stack storing the partial entity chunk, and β is a buffer holding unprocessed words. R is a set of relation arcs. E is a set of entity arcs. We use an index i to represent word wi and entity ei, respectively. A is used to store the action history.

Transitions	Change of State
LEFT$_l$-REDUCE	$\dfrac{([\sigma \mid i^*],\ \delta, e,\ [j^* \mid \beta],\ R, E)}{(\sigma,\ \delta, e,\ [j^* \mid \beta],\ R \cup \{(i^* \xleftarrow{l} j^*)\}, E)}$
RIGHT$_l$-SHIFT	$\dfrac{([\sigma \mid i^*],\ \delta, e,\ [j^* \mid \beta],\ R, E)}{([\sigma \mid i^* \mid \delta \mid j^*],\ [\,], e,\ \beta,\ R \cup \{(i^* \xrightarrow{l} j^*)\}, E)}$
NO-SHIFT	$\dfrac{([\sigma \mid i^*],\ \delta, e,\ [j^* \mid \beta],\ R, E)}{([\sigma \mid i^* \mid \delta \mid j^*],\ [\,], e,\ \beta,\ R, E)}$
NO-REDUCE	$\dfrac{([\sigma \mid i^*],\ \delta, e,\ [j^* \mid \beta],\ R, E)}{(\sigma,\ \delta, e,\ [j^* \mid \beta],\ R, E)}$
LEFT$_l$-PASS	$\dfrac{([\sigma \mid i^*],\ \delta, e,\ [j^* \mid \beta],\ R, E)}{(\sigma,\ [i^* \mid \delta], e,\ [j^* \mid \beta],\ R \cup \{(i^* \xleftarrow{l} j^*)\}, E)}$
RIGHT$_l$-PASS	$\dfrac{([\sigma \mid i^*],\ \delta, e,\ [j^* \mid \beta],\ R, E)}{(\sigma,\ [i^* \mid \delta], e,\ [j^* \mid \beta],\ R \cup \{(i^* \xrightarrow{l} j^*)\}, E)}$
NO-PASS	$\dfrac{([\sigma \mid i^*],\ \delta, e,\ [j^* \mid \beta],\ R, E)}{(\sigma,\ [i^* \mid \delta], e,\ [j^* \mid \beta],\ R, E)}$

O-DELETE	$\dfrac{([\sigma\|i^*],\ \delta, e,\ [j\|\beta],\ R, E)}{([\sigma\|i^*],\ \delta, e,\ \beta,\ R, E)}$
GEN-SHIFT	$\dfrac{([\sigma\|i^*],\ \delta, e,\ [j\|\beta],\ R, E)}{([\sigma\|i^*],\ \delta,\ [j\|e], \beta,\ R, E)}$
GEN-NER(y)	$\dfrac{([\sigma\|i^*],\ \delta, [j\|e],\ [\beta],\ R, E)}{([\sigma\|i^*],\ \delta, [\ ],\ [j^*\|\beta],\ R,\ E \cup \{j^*\})}$



State	Transition	σ	δ	e	β	R	E
0	Initialization	[ ]	[ ]	[ ]	$[1, \ldots, 6]$	$\emptyset$	
1	GEN-SHIFT	[ ]	[ ]	[1]	$[2, \ldots, 6]$		
2	GEN-NER	[ ]	[ ]	[ ]	$[1^*, \ldots, 6]$		$E \cup \{1 - \text{Per} \rightarrow 1\}$
3	NO-SHIFT	$[1^*]$	[ ]	[ ]	$[2, \ldots, 6]$		
4	O-DELETE	$[1^*]$	[ ]	[ ]	$[3, \ldots, 6]$		
5	O-DELETE	$[1^*]$	[ ]	[ ]	$[4, \ldots, 6]$		
6	GEN-SHIFT	$[1^*]$	[ ]	[4]	$[5, 6]$		
7	GEN-SHIFT	$[1^*]$	[ ]	$[4, 5]$	$[6]$		
8	GEN-NER	$[1^*]$	[ ]	[ ]	$[5^*, 6]$		$E \cup \{4 - \text{Loc} \rightarrow 5\}$
9	RIGHT-SHIFT	$[1^*, 5^*]$	[ ]	[ ]	$[6]$	$R \cup \{1 - \text{Live\_In} \rightarrow 5\}$	
10	GEN-SHIFT	$[1^*, 5^*]$	[ ]	[6]	[]		
11	GEN-NER	$[1^*, 5^*]$	[ ]	[ ]	$[6^*]$		$E \cup \{6 - \text{Loc} \rightarrow 6\}$
12	RIGHT-PASS	$[1^*]$	$[5^*]$	[ ]	$[6^*]$	$R \cup \{5 - \text{Loc\_In} \rightarrow 6\}$	
13	RIGHT-SHIFT	$[1^*, 5^*, 6^*]$	[ ]	[ ]	[]	$R \cup \{1 - \text{Live\_In} \rightarrow 6\}$	

Table 3: Transition sequence for the entity and relation graph in Figure 1.

Input Representation

– The bottom layer is token embedding and the next layer is a Bi-LSTM layer to capture richer contextual information.

$$x_i = \max\{0, V[\widetilde{w}; w] + b\},$$

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}].$$

State Representation

- Stack LSTM [Dyer *et al.*, 2015] to represent different components of each state
- In a stack LSTM, the current location of a stack pointer determines which cell in the LSTM provides c_{t-1} and h_{t-1} when computing the new memory cell contents. The stack LSTM provides a *pop* operation which moves the stack pointer to the previous element. Thus, the stack-LSTM can be understood as a stack implemented so that contents are never overwritten.

Composition Functions

- Entity Chunks :When GEN-NER(y) is executed, the algorithm shifts the sequence of words on e to the top of β as a single completed chunk. To compute an embedding of this sequence, we run a bidirectional LSTM over the embeddings of its constituent words together with the chunk type (i.e., y).
- Relation labels: Given a directed relation arc, which points from a head node h to a modifier node m, we combine both head-modifier pair and modifier-head pair, and use the combinations to update the embeddings of head node and modifier node separately.

$$c = \tanh(W^h[H; M; R] + e^h) \qquad c = \tanh(W^t[M; H; R] + e^t)$$

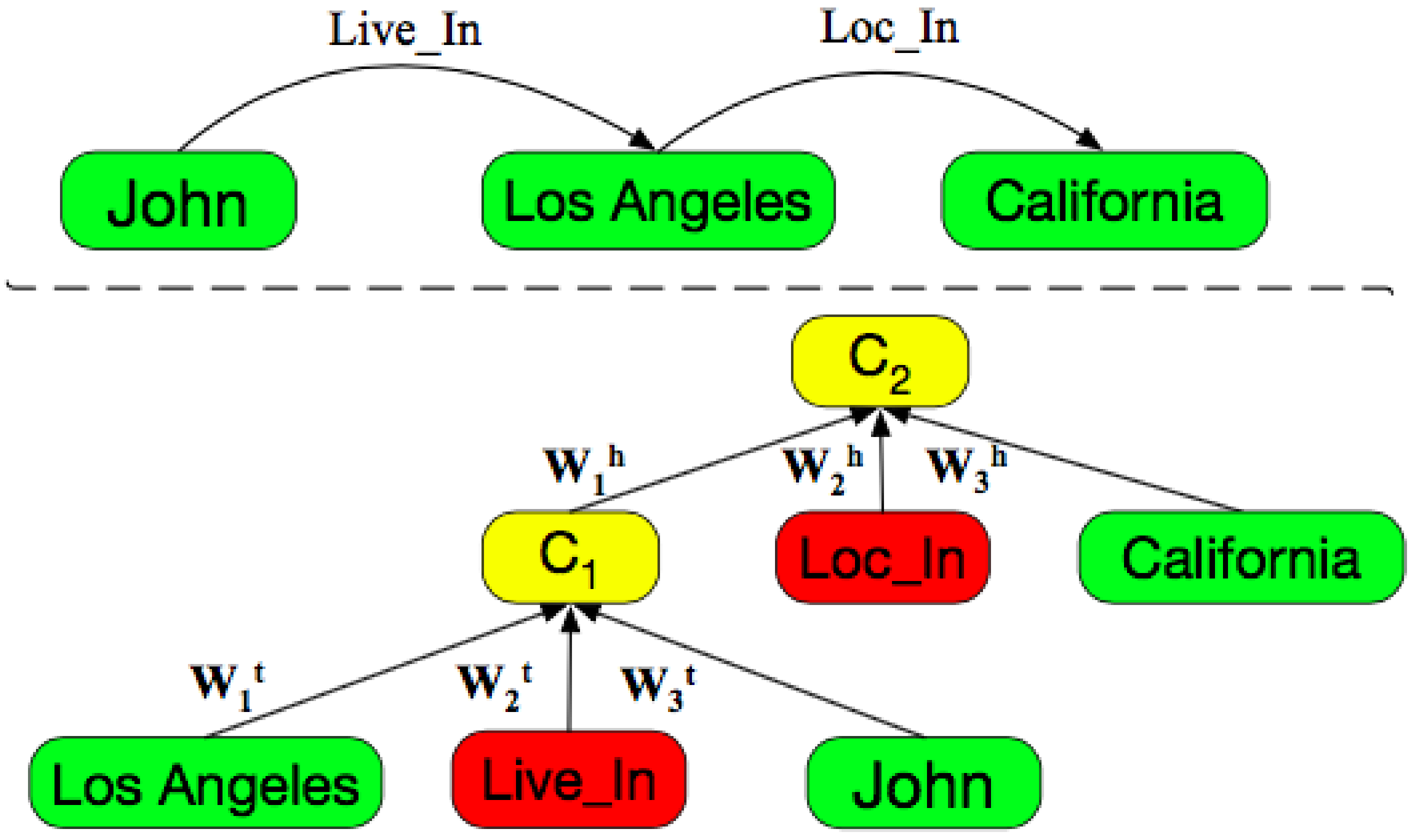


Figure 4: Relation representation of "*Los Angeles*", computed by recursively applying composition functions. W^t is the parameters when "*Los Angeles*" is the modifier and W^h is the parameters when "*Los Angeles*" is the head

Experiments

Method	Prec.	Rec.	F1
FCM [Gormley *et al.*, 2015]	55.3	15.4	24.0
DS+logistic [Mintz *et al.*, 2009]	25.8	39.3	31.1
LINE [Tang *et al.*, 2015]	33.5	32.9	33.2
MultiR [Hoffmann *et al.*, 2011]	33.8	32.7	33.3
DS-Joint [Li and Ji, 2014]	57.4	25.6	35.4
CoType [Ren *et al.*, 2017]	42.3	51.1	46.3
LSTM-LSTM-Bias	61.5	41.4	49.5
LSTM-LSTM-Bias*	60.8	41.3	49.1
Our Method	**64.3**	**42.1**	**50.9**

Table 4: Comparison with previous state-of-the-art methods on NYT. The first part (from row 1 to row 3) is the pipelined methods, the second part (row 4 to 6) is the jointly extracting methods, and the third part (row 7 to 9) is the end-to-end methods.

Standard S1:	CARVING THE GRANITE A Park as Eight Miles of Exits The Flume, [Cannon Mountain]$_{\text{LOC:CONTAIN-2}}$ and the Old Man of the Mountain Historic Site are all part of Franconia Notch State Park in [New Hampshire]$_{\text{LOC:CONTAIN-1}}$.
LSTM-LSTM-Bias*:	CARVING THE GRANITE A Park as Eight Miles of Exits The Flume, [Cannon Mountain] and the Old Man of the Mountain Historic Site are all part of Franconia Notch State Park in [New Hampshire].
Our Model:	CARVING THE GRANITE A Park as Eight Miles of Exits The Flume, [Cannon Mountain]$_{\text{LOC:CONTAIN-2}}$ and the Old Man of the Mountain Historic Site are all part of Franconia Notch State Park in [New Hampshire]$_{\text{LOC:CONTAIN-1}}$.
Standard S2:	The US offered to locate the missile system in Poland, drawing furious objections from [Russia]$_{\text{LOC-LAC-1}}$, though [Washington]$_{\text{LOC-LAC-1}}$ argues that the system is built to defend against [Iran]$_{\text{LOC-LAC-2,LOC-LAC-2}}$, principally.
LSTM-LSTM-Bias*:	The US offered to locate the missile system in Poland, drawing furious objections from [Russia]$_{\text{LOC-LAC-1}}$, though [Washington] argues that the system is built to defend against [Iran]$_{\text{LOC-LAC-2}}$, principally.
Our Model:	The US offered to locate the missile system in Poland, drawing furious objections from [Russia]$_{\text{LOC-LAC-1}}$, though [Washington]$_{\text{LOC-LAC-1}}$ argues that the system is built to defend against [Iran]$_{\text{LOC-LAC-2,LOC-LAC-2}}$, principally.

Table 6: Output from LSTM-LSTM-Bias and our model. The first row for each example is the gold standard. "*LOC*" is entity type, "*CONTAIN*" and "*LAC*" are relation types, "1" and "2" mean direction of relation. The color of "*LOC-LAC**" refers to relation instance.