Event Schema Induction Based on Relational Co-occurrence over Multiple Documents

Tingsong Jiang, Lei Sha, and Zhifang Sui

Key Laboratory of Computational Linguistics, Ministry of Education,

Institute of Computational Linguistics, School of Electronics and Engineering and Computer Science, Peking University

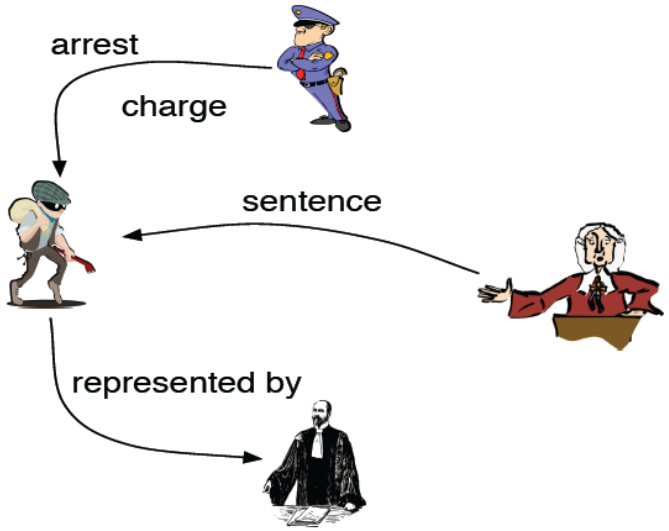
Beijing, China

北京大学

Outline

- Introduction
- Motivation
- Related Work
- System Overview
- Co-occurrence over Multiple Documents
- Co-occurrence Graph Analysis
- Experiments

Introduction



- Event schema is a template
  - a set of events + a set of slots or roles
- Event schemas specify actors and their roles within events
  - a bombing event: perpetrator, victim, and instrument
  - an arrest event: agent, police, suspect, judge
- Application: helpful for information extraction and other NLP tasks
- Traditional IE systems: design the template manually and focus on extracting structured information to fill predefined templates.
- Drawback:
  - limit event templates' range of application to a relatively narrow area
  - only applied in a particular domain

Motivation

- basic assumption: related events tend to appear together to describe a scenario in natural-language discourse
- Related events co-occurrence in one document
- Related events co-occurrence over more than 2 documents
- relational tuples co-occurrence over multiple documents may help to combine some loose, scattered and relative event schemas
- relational tuples co-occurrence over multiple documents may also help alleviate incoherence problems

Example

- a person returning to his work
- 3 schemas are somehow similar
- scientist Schwarzschild

and his life in war

- Both (Schwarzschild,

return in, United States Army)

→combine the event schemas

Schema 1:				
A1:[person]	return to	A0:	[none]	lineup
A1:[person]	return against	A3:[location;organization]		
A1:[person]	return for	A4:[activity;game]		
A1:[person]	return after	A6:	[none]	absence
A1:[person]	**return in**	**A8:[location] United States Army**		
A1:[person]	return until	A9:[time_unit]		
Schema 2:				
A0:[person]	return as	A1:[person]		
A0:[person]	return after	A5:	[none]	year
A0:[person]	return to	A9:[time_period]		
A0:[person]	be survive by	A10:	[none]wife	
A0:[person]	**return in**	**A11:[location] United States Army**		
Schema 3:				
A0:[person]	leave for	A1:[location] Europe		
A0:[person]	leave with	A6:[person]		
A0:[person]	leave to meet with	A8:[person]		
A0:[person]	leave after	A9:	[none]	season
A0:[person]	leave on	A11:[organization]		
A0:[person]	leave as	A12:[leader]		

Related Work

- Event template extraction has been explored in the MUC-4 scenario template task. Still needs large-scale event schema extraction system.
- Automatically induced and large-scale: the seminal work of Chambers and Jurafsky (2008,2009,2011)– use subject-verb and verb-object representation: A1 caused A2; A2 burned A1
- Niranjan (2013) extended the system using relational n-grams--use relational subject-verb-object tuples

System overview

- Both frequently co-occurring relational tuples in one document and over different documents can reflect the relatedness of assertions about real-world events
- Preprocessing:
- Use Open Information Extraction and extract the relational tuples of the form (Arg1 , Relation , Arg2)
- Generalized by semantic types from WordNet and Stanford Named Entity Recognizer

Preprocessing

- “Woz and Jobs started Apple in my parents’ garage in 1976.”
- divided into tuples as:
  - 1.(Wos and Jobs, started, Apple)
  - 2.(Wos and Jobs, started in, my parents’ garage)
  - 3.(Wos and Jobs, started in, 1976)
- generalized as follows:
  - 1.(Wos and Jobs, started, Apple)
  - 2.(<person>, start, <org>)
  - 3.(<person>, start in, <location>)
  - 4.(<person>, start in, <time_unit>)

Relational Tuples Co-occurrence over Multiple Documents

- Co-occurrence in one document can be defined as the count that they appear in a window.
- co-occurrence over multiple documents may show transitivity
  - Document 1:X follows Y; Document 2:Z follows Y
  - Both weighted co-occurrences reach the threshold
  - X and Z are relational somehow
- define the quality of co-occurrence as

$$C(x,y)=\frac{1}{2}\left[f(x,y)/\ f(x)+ f(x,y)/\ f(y)+g(x,y)/[\ f(x)+ f(y)\right]$$

$$d(x,y)=e^{-\gamma(k-1)}$$

$$C'(x,y)=C(x,y)\cdot d(x,y)$$

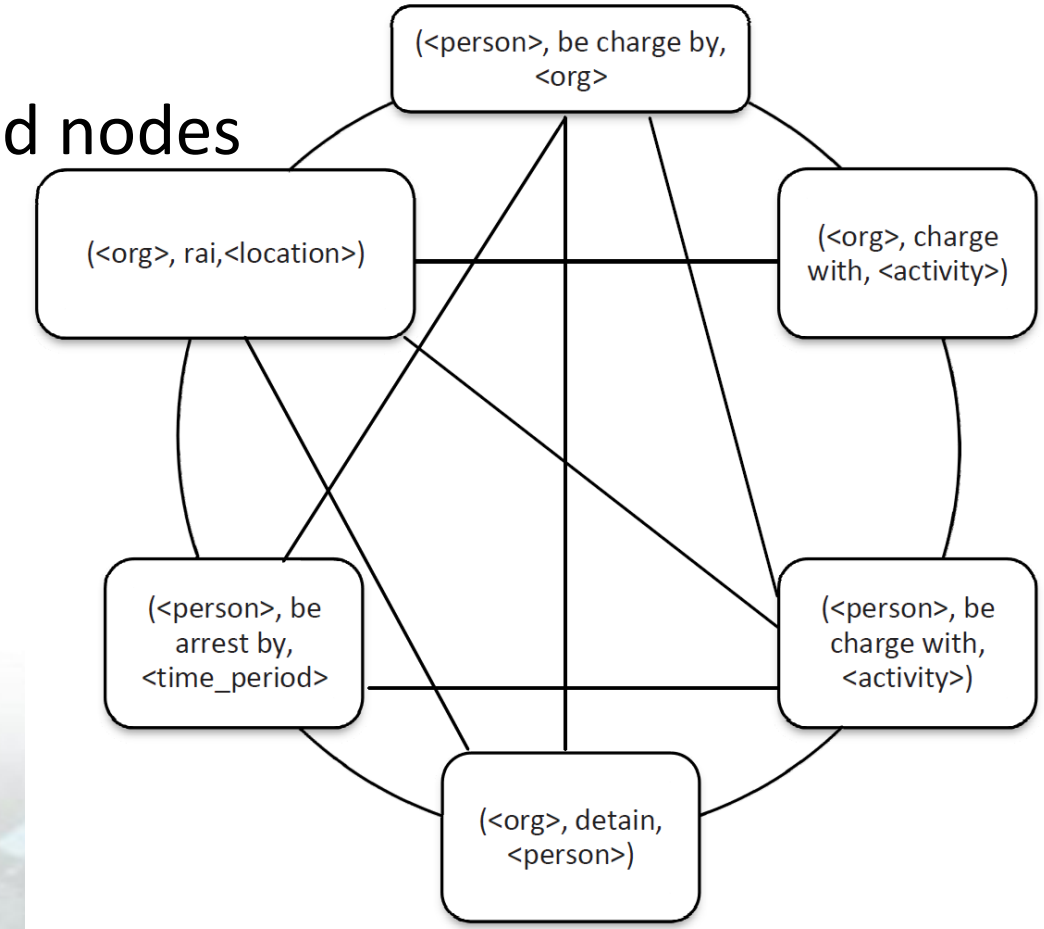
id	Arg1	Relation	Arg2	Count
…	…	…	…	…
23	Bomb	Explode in	<location>	547
24	Bomb	Explode in	Baghdad	22
25	Bomb	Explode in	Market	7
…	…	…	…	…
69	Bomb	Kill	<person>	173
…	…	…	…	…
95	<person>	Be identified	<org>	231
…	…	…	…	…

X	Y	flag	Distance	Count	E11	E12	E21	E22
…	…	…	…	…	…	…	…	…
23	69	0	1	27	25	0	0	0
23	69	0	2	35	33	0	0	0
…	…	…	…	…	…	…	…	…
23	69	0	10	62	59	0	0	0
69	23	0	1	6	0	0	0	0
…	…	…	…	…	…	…	…	…
69	95	0	1	18	0	0	32	0
69	95	0	2	12	0	0	9	0
…	…	…	…	…	…	…	…	…
69	95	0	10	54	0	0	50	0
95	69	0	1	14	0	36	0	0
…	…	…	…	…	…	…	…	…
23	95	1	3	46	0	0	0	0
23	95	1	4	39	0	0	0	0
…	…	…	…	…	…	..	…	…

- Doc A:
  - id23=(bomb, explode in, <location>)
  - id69=(bomb, kill,<person>)
  - Co-occur in distance=1 &27 times
- Doc B:
  - id69=(bomb, kill, <person>) ,
  - id71=(<person>, be sent to , hospital),
  - id95=(<person>, be identified , perpetrator),
- Flag=0 co-occurrence in one doc
- Flag=1 co-occurrence in multiple docs
- A1_B2: 27+12=39; A2_B1: 35+18=53; A_B_d3: 39+53=92; normalized: 92/2=46
  - Total= A1_B2+A2_B1+A1_C2+A2_C1… normalized: total/3…
- E11 means X.Arg1=Y.Arg1, E12 means X.Arg1=Y.Arg2 and so on.

Co-occurrence Graph Analysis

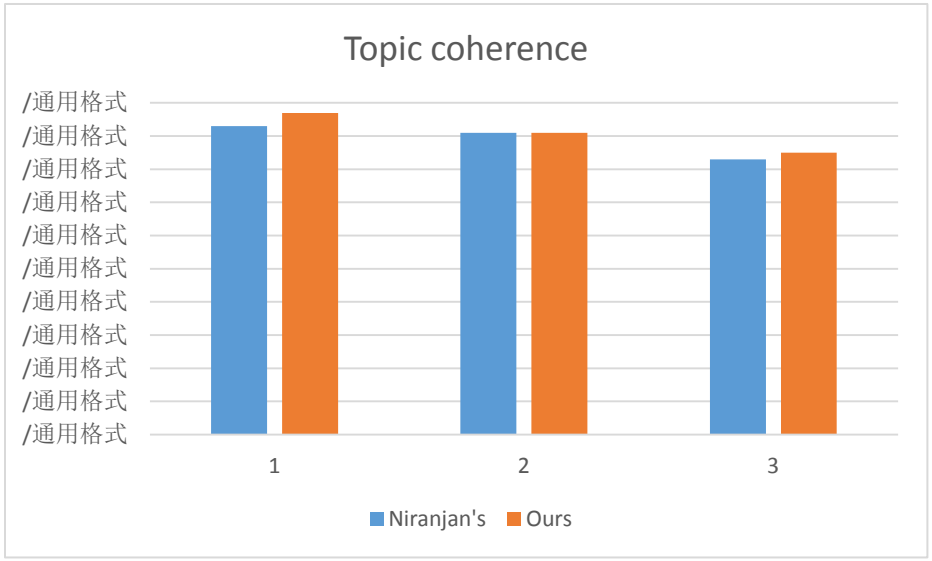
- G=(V,E) whose nodes are relation tuples and edges are weighted co-occurrence value C'(x,y)
- graph analysis to find the most related nodes
- use them to create an event schema
- graph analysis: Page rank algorithm
- event schema generation
  - a ranking list of tuples
  - the top n elements to be event schema

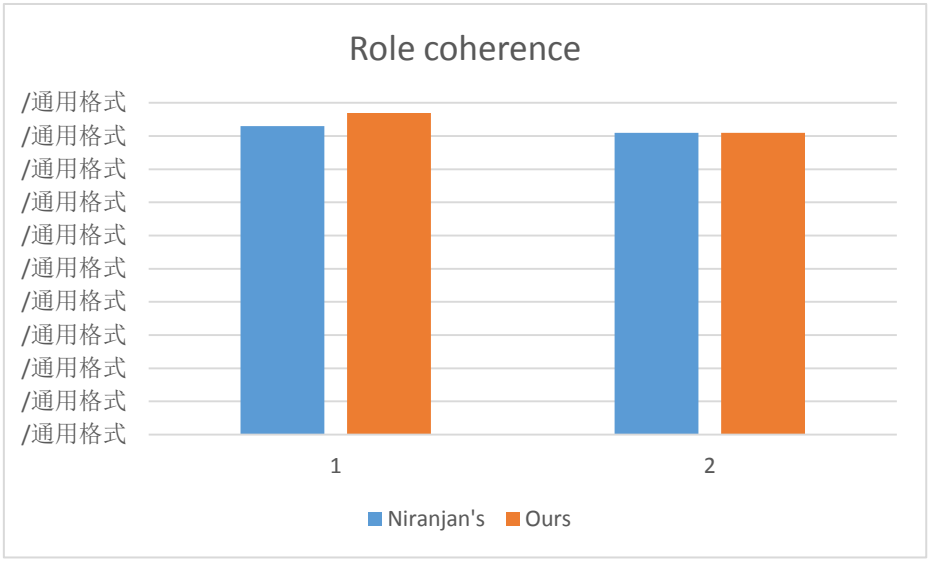


Experiments

- Find whether the schema generated has coherence of real world
- *Topic coherence:* aimed to test whether the relations in a schema form a coherent topic or event
- *Role Coherence*: aimed to test whether the instances of a role form a coherent set.
- sample some the output schemas and examine whether they make sense in the real world by humans

Results





- Two evaluation tasks are performed to test the coherence and validity of the event schema and the roles.
- We found that 96% of the schemas have sense in the real world, and 94% of the roles are reasonable and not mixed up in one schema

北京大学

- compare our event schemas with the MUC-4 templates
- A proportion of slots discovered for each MUC-4 terrorist event template
- Why? relational tuples co-occurrence over multiple documents was helpful and assisted the graph analysis to extract and merge the bombing event element.

Template	perpetrator	victim	Physical target	instrument	location	date
bombing	0.935	0.923	0.912	0.931	0.957	0.953
attack	0.928	0.939	0.914	0.945	0.967	0.962
kidnapping	0.892	0.921	N.A.	N.A.	0.630	0.840
arson	0.921	0.933	0.870	N.A.	0.920	0.933

Results

- Relational tuples co-occurrence over multiple documents can add the weight between nodes and graph analysis can find the most relational nodes correctly.
- As a result, some event schemas may be richer and more coherent in our real world.
- Co-occurrence may link some roles over multiple documents and the role candidates are more complete to be induced.

Conclusion

- exploit the relational tuples co-occurrence both in one document and over multiple relational documents
- build a relational graph with the nodes of the form (Arg1, relation, Arg2).
- Relational tuples co-occurrence over multiple documents may help to simplify the event schema
- relational tuples co-occurrence over multiple documents can find more instances for roles in event schema which may enrich the event and make it more coherent in the real world.
- Future: investigate event schema induction evaluation, for example to evaluate event coherence automatically in addition to the slots and entities

Thanks!

北京大学