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
{tingsong,shalei,szf}@pku.edu.cn

Abstract. Event schema which comprises a set of related events and participants is of great importance with the development of information extraction (IE) and inducing event schema is prerequisite for IE and natural language generation. Event schema and slots are usually designed manually for traditional IE tasks. Methods for inducing event schemas automatically have been proposed recently. One of the fundamental assumptions in event schema induction is that related events tend to appear together to describe a scenario in natural-language discourse, meanwhile previous work only focused on co-occurrence in one document. We find that semantically typed relational tuples co-occurrence over multiple documents is helpful to construct event schema. We exploit the relational tuples co-occurrence over multiple documents by locating the key tuple and counting relational tuples, and build a co-occurrence graph which takes account of co-occurrence information over multiple documents. Experiments show that co-occurrence information over multiple documents can help to combine similar elements of event schema as well as to alleviate incoherence problems.

Keywords: event schema, information extraction, co-occurrence analysis.

1 Introduction

Event schema is a template which comprises a set of events with prototypical transitions as well as a set of slots or roles representing the participants. Various roles or slots are contained in one event, such as the perpetrator, victim, and instrument in a bombing event. Event schema is helpful for information extraction and other NLP tasks due to the fact that both abstraction and concreteness are necessary information for people to understand the event. Traditional IE systems design the template manually and focus on extracting structured information concerning events to fill predefined templates. This approach often limits event templates' range of application to a relatively narrow area and is only applied in a particular domain.

C. Zong et al. (Eds.): NLPCC 2014, CCIS 496, pp. 23–33, 2014.

One of the basic assumptions in event schema induction is that related events tend to appear together to describe a scenario in natural-language discourse, meanwhile previous work only focused on co-occurrence in one document. Besides, using relational tuples and generalization also makes the schema loose and scattered. We find that semantically typed relational tuples co-occurrence over multiple documents is helpful to construct event schema.

First, relational tuples co-occurrence over multiple documents may help to combine some loose and relative event schemas. Some event schemas may be scattered due to their lack of enough co-occurrence, such as the events shown in Figure 1. The event schemas in Figure 1 make some assertions that a person returning to his work but the schemas are somehow similar and describe the event parallelly which seems not that simple. If we exploit the relational co-occurrence over multiple documents, we can find that schema 1 in document A and schema 2 in document B have something in common where they tell us a person returned to some location at some time. We notice that it is induced from a document that describes the scientist Schwarzschild and his life in war, both schema 1 and schema 2 have the tuple that (Schwarzschild, return in, United States Army), and if we find the co-occurrence in two documents and count the relational tuples again, we can combine the event schemas and construct a relatively more abstract and complete event schema.

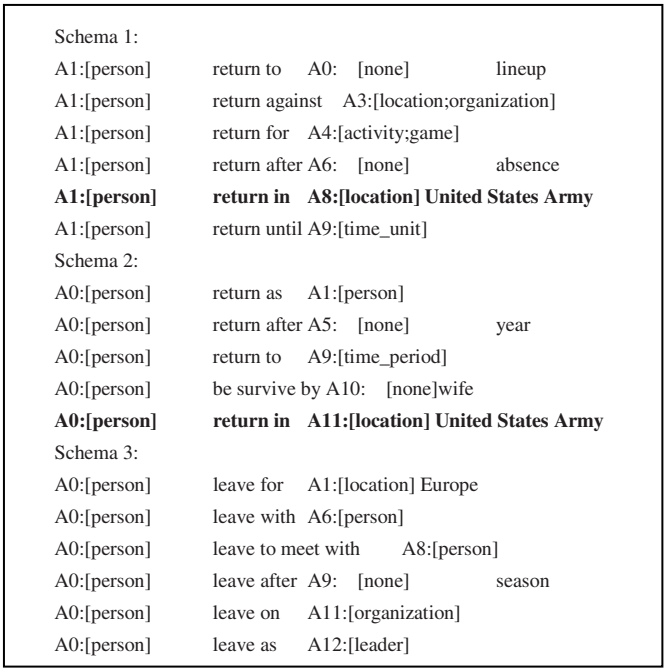

Fig. 1. An example of Niranjan, Stephen and Mausam and Oren(2013)'s system. Schema 1~3 are similar and describe the same event that one scientist returned to his work or location and can be combined and integrated.

Furthermore, we find in the experiment that relational tuples co-occurrence over multiple documents may also help alleviate incoherence problems. Chamber's system lack coherence because of the representation that uses pairs of elements from an assertion, thus, treating subject-verb and verb-object separately, and in a result their system may mix unrelated events and have roles whose entities do not play the same role in the schema. For example Niranjan pointed out Chamber's system mixed the events of fire spreading and disease spreading in Niranjan, Stephen and Mausam and Oren(2013), and they used relational tuples instead. However, we find that incoherence problems still exists.

In this paper we propose a method for event schema induction based on relational co-occurrence over multiple documents which overcomes scatter problems and incoherence in event schema induction. The rest of the paper is organized as follows: in Section 2 we review the related work while Section 3 presents a general overview of our approach and how relational co-occurrence over multiple documents is obtained and used in event schema induction. Finally, Section 4 gives the results of our approach.

2 Related Work

Since the late 1980s, information extraction began to flourish, this is mainly attributed to Message Understanding Conference (MUC) and Automatic Content Extraction (ACE) meeting. Event template extraction has been explored in the MUC-4 scenario template task. This task concentrated on pipeline models which decouple the task into the sub-tasks of field extraction and event-based text segmentation. The application of the current event schema generation and other semantic processing tasks is mainly by manual effort to define the target representation and annotate the examples to train the machine-learning system, besides MUC template is confined to specific areas, so large-scale event schema extraction system is desperately needed.

Event schemas in NLP were not automatically induced until the seminal work of Chambers and Jurafsky(2009). Their work is fully automatical and can deal with large text corpora. Early work by Chambers and Jurafsky (2008; 2009) showed that we could exploit the event sequences from text to induce event schemas automatically. They used pair-wise subject-verb and verb-object representation which may result to incoherence in the real world. Chambers and Jurafsky (2011) had applied unsupervised techniques to combine clustering, semantic roles, and syntactic relations so as to construct templates as well as to fill slots. Niranjan, Stephen, Mausam and Oren (2013) extended the system using relational n-grams to find the co-occurrence statistics of relational subject-verb-object tuples instead of pair-wise subject-verb/verb-object representation, and they aimed to overcome incoherence problems. However, Niranjan's system only considered the co-occurrence statistics in one document which may omit relations and information over multiple documents.

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

In this section, we show the event schema induction process and deal with acquisition of the relational tuples co-occurrence over multiple documents.

3.1 System Overview

The overall system is based on the idea that both frequently co-occurring relational tuples in one document and over different documents can reflect the relatedness of assertions about real-world events. The preprocessing includes extracting the relational tuples from the corpus and counting the co-occurrence of relational tuples. The relational tuples can be treated as the nodes of a graph, and we can apply graph analysis to find the most relational nodes and then we can cluster them as part of the arguments of the event schema.

Event schema induction of Chamber's work is focused on the verbs and their attribute thus verbs are the main features that distinguish one event from others according to the relations between verbs and events. Niranjan, Stephen, Mausam and Oren (2013) proposed that relational tuples is more expressive. We use Open Information Extraction (Mausam et al.,2012) and extract the relational tuples of the form (Arg1 , Relation , Arg2) which may contain more coherence than pair-wise representation. To reduce the sparsity resulted from tuples we use head verb and preposition to describe a phrase.

To ensure the tuples are not overly specific it can be generalized by semantic types from WordNet(Niranjan, Stephen, Mausam and Oren(2013)). The set of types are: person, organization, location, time unit, number, amount, group, business, executive, leader, effect, activity, game, sport, device, equipment, structure, building, substance, nutrient, drug, illness, organ, animal, bird, fish, art, book, and publication. We use Stanford Named Entity Recognizer (Finkel et al., 2005), and also look up the argument in WordNet 2.1 and record the first three senses if they map to the target semantic types. The sentence "Woz and Jobs started Apple in my parents' garage in 1976." could be divided into tuples as:

1.(Wos and Jobs, started, Apple)
2.(Wos and Jobs, started in, my parents' grage)
3.(Wos and Jobs, started in, 1976)
Then it is generalized as follows:
1.(Wos and Jobs, started, Apple)
2.(<person>, start, <org>)
3.(<person>, start in, <location>)
4.(<person>, start in, <time_unit>)
...

After relation extraction and representation, we calculate the co-occurrence tuples both in one document and over multiple documents as detailed in Section 3.2. In Section 3.3 we construct a relational graph using the relational tuples as nodes and

co-occurrence quality as weighed edges, and apply graph analysis to find the key elements for event schema.

3.2 Relational Tuples Co-occurrence over Multiple Documents

As mentioned above, considering co-occurrence over multiple documents may help to combine event elements and alleviate incoherence problem in event schema induction. In the corpus, some documents may refer to the same topic or implicate the relations, others just have nothing to do with it. Since the text corpus may be large, we should first cluster the documents according to their types and features, then we find the common tuples over two documents and count the co-occurrence.

3.2.1 An Acquisition Method for Relational Tuples Co-occurrence over Multiple Documents Based on Transitivity

The basic intuition is that most frequent co-occurrence of two tuples reflect tight connection in the real world. Co-occurrence in one document can be defined easily as the count that they appear in a window. But co-occurrence over multiple documents may be not that clear and easy. We find that co-occurrence over multiple documents may show transitivity. We consider the example of a bombing event. In the first news article, we calculate the co-occurrence number and find tuple X: (bomb, explode, <location>) and tuple Y: (bomb, kill, <person>) are highly connected. In the second news article we calculate tuple Y: (bomb, kill, <person>) and tuple Z: (<person>, be identified, perpetrator) are frequently appeared. We can infer that X and Z have a latent relation and we thus try to find the co-occurrence statistics between them.

If we find that X follows Y and the weighted co-occurrence reaches the threshold in one document, besides, we find that Z follows Y and also the weighted co-occurrence reaches the threshold. Then we can guess X and Z are relational somehow, and we should count the co-occurrence between them by comparing the two documents. Specifically, we should first find the same tuple that joins the two documents together, and then we add the distance together for one occurrence. By counting the co-occurrence number with their distances like this, a new "co-occurrence count" parameter is obtained. The database thus can be shown in Figure 2.Suppose in document A: a tuple list contains: id23=(bomb, explode in, <location>) , id69=(bomb, kill, <person>), they occur once in distance=1 and they occur 27 times in the overall documents; In document B: the tuple list is in time order, i.e. id69= (bomb, kill, <person >) , id71=(<person>, be sent to , hospital), id95= (<person>, be identified , perpetrator), id69 and id95 occur once in distance=2; then we consider the co-occurrence over these two documents, and find they occur once in distance=3. Likewise, we then count the co-occurrence of (id23, id95) in distance=3 over the cluster of documents and add up them to the table and get Count=47. Note that the "Count=47" is normalized to be fair to traditional co-occurrence in one document.

Table 1.

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
…	…	…	…	…

Table 2.

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	47	0	0	0	0
23	95	1	4	39	0	0	0	0
…	…	…	…	…	…	..	…	…

Fig. 2. Table A represents the basic statistics of relational tuples count occur over the documents. Table B represents the co-occurrence statistics of relational tuples both in one documents and over multiple documents within different distances. Tag "flag=1" represents co-occurrence over multiple documents and "flag=0" means co-occurrence in each document and we add up each document's statistics to fill the table. Parameter E11 means X.Arg1=Y.Arg1, E12 means X.Arg1=Y.Arg2 and so on.

3.2.2 Co-occurrence Calculation

When we consider the co-occurrence both in one document and over multiple documents, we define the quality of co-occurrence as

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

where *f(x,y)* is a count for two tuples occurring in the same document, *f(x)* is the total count that x occurs ineach document; *f(y)* is the total count that tuple y occurs in document; *g(x,y)* is the co-occurrence count over the documents for x and y which is normalized due to the permutation and combination. We consider the distance which may influence the co-occurrence quality. Suppose (x,y) is a pair of tuples that occur in one document, the co-occurrence distance factor is defined as

$$d(x, y) = e^{-\gamma(k-1)} \tag{2}$$

where γ is the revision parameter, usually take the value of 0.5. The co-occurrence value is thus represented as

$$C'(x, y) = C(x, y) \cdot d(x, y) \tag{3}$$

3.3 Co-occurrence Graph Analysis

We construct a co-occurrence graph G=(V,E) whose nodes are relation tuples and edges are weighted co-occurrence value C'(x,y). Figure.3 shows a co-occurrence graph. If $C'(x, y) \geq \lambda$,where λ is the threshold, then link the x node with node y.

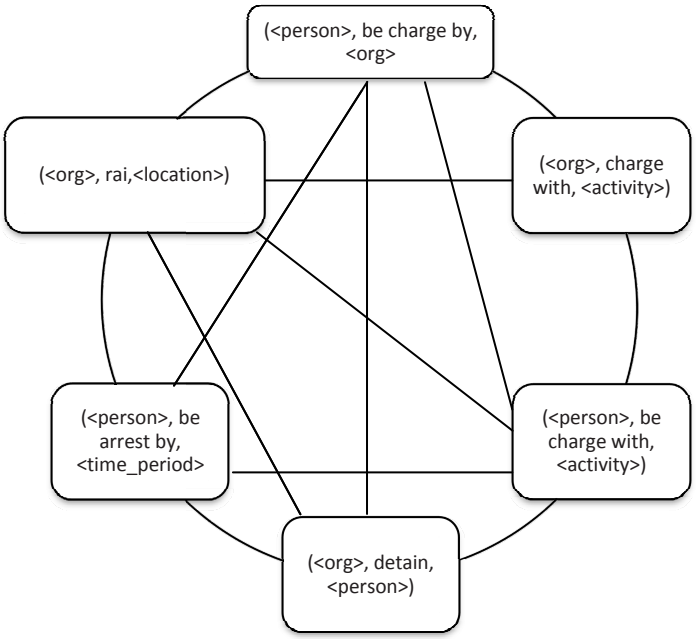

Fig. 3. Part of a graph showing tuples strongly associated with (<person>, be charge by, <org>)

For each cluster of co-occurrence graph, we perform graph analysis to find the most related nodes and use them to create an event schema. Page rank algorithm can be well adapted to our graph analysis. We use the Stanford Personalized PageRank algorithm[1] to rank the graph and obtained the top n nodes from the graph which may establish an event schema. For the top ranking node tuple X: (Arg1, *Rel*, Arg2), we record two roles (R1, R2) as the roles of the schema and use *Rel* to be the relation they have in the event. Then, we merge similar roles into one role according to Niranjan and Oren(2013).

Finally, the event schema generation is done with a ranking list of tuples. And we take the top n elements to be represented as part of the event schema.

4 Experiments

In this experiment we are desired to know whether the schema generated by our method has coherence of real world, such as common topic and correct roles. Since the output is large due to the large-scale corpus, we sample some the output schemas and examine whether they make sense in the real world.

4.1 Evaluation Methods

Is there an underlying common topic or event among the elements of the event schema? We focus on two measures, *topic coherence* and *role coherence* as mentioned in Niranjan and Oren(2013). A good schema must be topically coherent, i.e.,the relations and roles should relate to some realworld topic or event. The tuples that comprise aschema should be valid assertions that make sensein the real world. Finally, each role in the schema should belong to a cohesive set that plays a consistent role in the relations. We compare event schemas considered relational tuples co-occurrence over multiple documents against schemas released byNiranjan[2].

Topic coherence is aimed to test whether the relations in a schema form a coherent topic or event,we presented the annotators with a schema asa set of grounded tuples, showing each relation inthe schema, but randomly selecting one of the top ninstances from each role. We collected five instantiations like this for every schema. Three questions are ready for annotators: (1) is each of the grounded tuples meaningful in the real world; (2) do the majority ofrelations form a coherent topic; and (3) does eachtuple belong to the common topic.

Role Coherence is aimed to test whether the instances of anrole form a coherent set. We held therelation and one role fixed and presented theannotators with the top 5 instances for the other role. For each event schema's element (Arg1, Relation, Arg2), for example, we fixed the Relation and Arg1 and tested the top n instances of Arg2 to ask the annotators whether it belongs to the topic.

[1] Available on `http://nlp.stanford.edu/projects/pagerank.shtml`
[2] Available at `http://relgrams.cs.washington.edu`

We are also interesting to compare our event schemas with the MUC-4 templates. One of the MUC-4 tasks is to extract information aboutterrorist events, such as the names of perpetrators, victims, instruments, etc.MUC-4 templates for terrorist events include bombing, attack, kidnapping, and arson. Each template has six slots: perpetrator, victim, physicaltarget (omitted for kidnapping), instrument (omittedfor kidnapping and arson), location and date. We picked out the event schemas that include the key words of the tuple: (bomb, explode at, <location>) (bomb, explode on, <time_unit>) (<person>, plant, bomb) (bomb, wound/kill, <person>) to describe the bombing template. We checked the roles of the schema as the comparison to MUC-4 template slots.

4.2 Results

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. Two evaluation tasks are performed to test the coherence and validity of the event schema and the roles.

We obtained a test set of 10000 schemas per system by randomly sampling from each system. We evaluated this test set by manual testing whether the relations in one schema show a coherent topic or event. We took an average of ratings from five annotators as the final annotation. Figure 4 and Figure 5 show us the topic coherence and role coherence respectively.

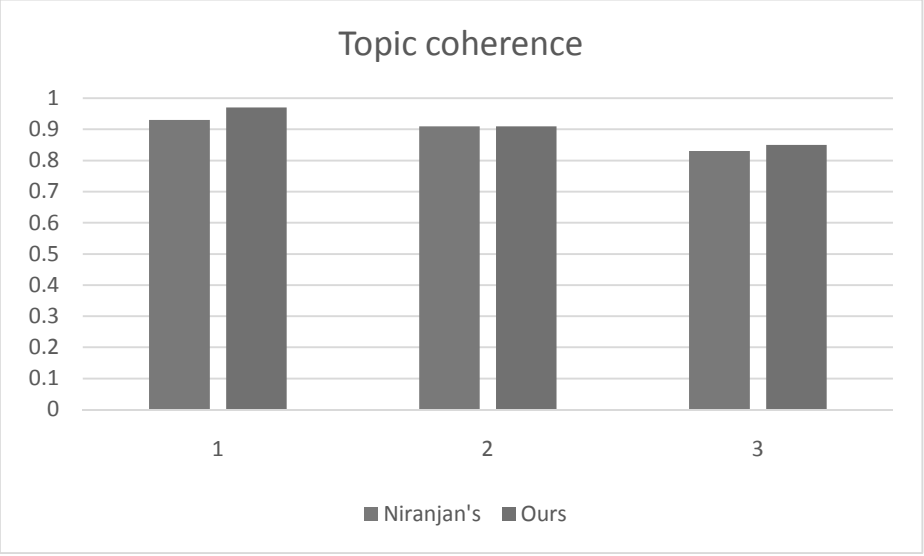

Fig. 4. Topic coherence bar gram shows whether the topic makes sense in the real-world. Histogram 1: percentage of schema instantiations with a coherent topic; Histogram 2: percentage of grounded valid tuple that assert valid real-world relations; Histogram 3: percentage of grounded statements where the instantiation has a coherent topic and the tuple is valid.

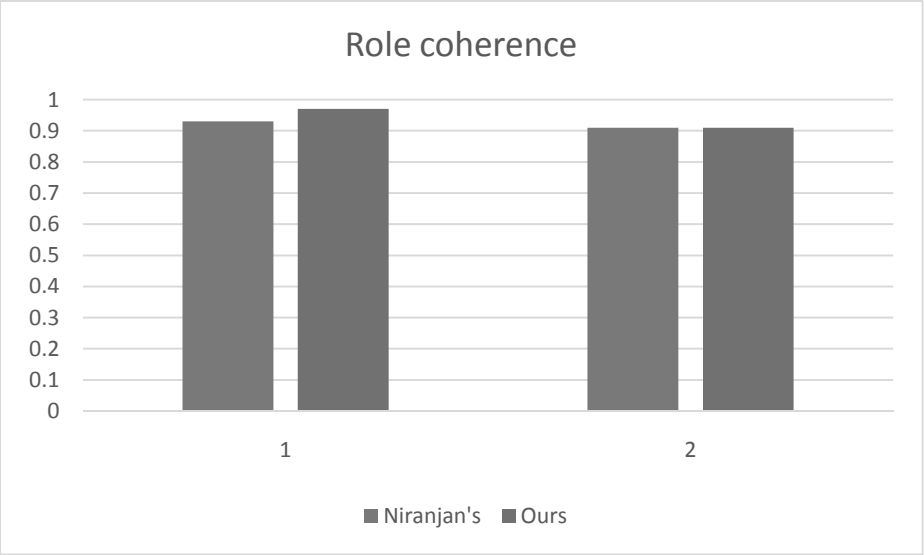

Fig. 5. Role coherence bar gram shows the roles in event schema correspond to common sense or not. Histogram 1: the percentage of tested roles of event schema which are coherent. Histogram 2: the percentage of top role instances which make sense in real-world.

Topic coherence is shown by topic coherence bars in Figure 3. Our result has a little advantage compared to Niranjan's. 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. Role coherence bars in Figure 4 shows the instances of roles that corresponding to the slots of the schema. Co-occurrence may link some roles over multiple documents and the role candidates are more complete to be induced.

The result compared with MUC-4 templates is shown in Table 1. A proportion of slots correctly discovered for each MUC-4 terrorist event template is represented. We can see that bombing event reflect the correspondence with MUC-4 template slots, as six slots are available at high proportion. We examined the corpus and the co-occurrence graph and found that relational tuples co-occurrence over multiple documents was helpful and assisted the graph analysis to extract and merge the bombing event element. Kidnapping event is weaker to discover the location slot as the co-occurrence both in one document and over multiple documents is not that tight.

Table 3. A proportion of slots discovered for each MUC-4 terrorist event template

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

5 Conclusion

We exploit the relational tuples co-occurrence both in one document and over multiple relational documents and 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 as pointed in Section 1. Besides, 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. We would like to investigate event schema induction evaluation, for example to evaluate event coherence automatically in addition to the slots and entities.

Acknowledgement. This paper is supported by National Key Basic Research Program of China 2014CB340504 and NSFC project 61375074.

References

1. Chambers, N., Jurafsky, D.: Unsupervisedlearning of narrative event chains. In: Proceedings of ACL 2008: HLT (2008)
2. Chambers, N., Jurafsky, D.: Unsupervisedlearning of narrative schemas and their participants. In: Proceedings of ACL (2009)
3. Chambers, N., Jurafsky, D.: Template-basedinformation extraction without the templates. In: Proceedings of ACL (2011)
4. Balasubramanian, N., Soderland, S., Mausam, Etzioni, O.: Generating Coherent Event Schemas at Scale. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013)
5. Balasubramanian, N., Soderland, S., Mausam, Etzioni, O.: Rel-grams: A Probabilistic Model of Relations in Text. In: Proc. of the Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extraction (AKBC-WEKEX) (2012)
6. Cheung, J., Poon, H., Vandervende, L.: Probabilistic frame induction. In: Proceedings of NAACL HLT (2013)
7. Barzilay, R.R.: Multi Event Extraction Guided by Global Constraints. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2012)