Joint Learning Templates and Slots for Event Schema Induction

Lei Sha, Sujian Li, Baobao Chang, Zhifang Sui
Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University
Collaborative Innovation Center for Language Ability, Xuzhou 221009 China
shalei, lisujian, chbb, szf@pku.edu.cn

Abstract

Automatic event schema induction (*AESI*) means to extract meta-event from raw text, in other words, to find out what types (templates) of event may exist in the raw text and what roles (slots) may exist in each event type. In this paper, we propose a joint entity-driven model to learn templates and slots simultaneously based on the constraints of templates and slots in the same sentence. In addition, the entities' semantic information is also considered for the inner connectivity of the entities. We borrow the *normalized cut* criteria in image segmentation to divide the entities into more accurate template clusters and slot clusters. The experiment shows that our model gains a relatively higher result than previous work.

1 Introduction

Event schema is a high-level representation of a bunch of similar events. It is very useful for the traditional information extraction (*IE*)(Sagayam et al., 2012) task. An example of event schema is shown in Table 1. Given the bombing schema, we only need to find proper words to fill the slots when extracting a bombing event.

There are two main approaches for *AESI* task. Both of them use the idea of clustering the potential event arguments to find the event schema. One of them is probabilistic graphical model (Chambers, 2013; Cheung, 2013). By incorporating templates and slots as latent topics, probabilistic graphical models learns those templates and slots that best explains the text. However, the graphical models considers the entities independently and do not take the interrelationship between entities into account. Another method relies on ad-hoc clustering algorithms (Filatova et al., 2006; Sekine, 2006; Chambers and Jurafsky, 2011). (Chambers and Jurafsky, 2011) is a pipelined approach. In the first step, it uses pointwise mutual information(PMI) between any two clauses in the same document to learn events, and then learns syntactic patterns as fillers. However, the pipelined approach suffers from the error propagation problem, which means the errors in the template clustering can lead to more errors in the slot clustering.

Bombing Template	
Perpetrator:	person
Victim:	person
Target:	public
Instrument:	bomb

Table 1: The event schema of bombing event in MUC-4, it has a bombing template and four main slots

This paper proposes an entity-driven model which jointly learns templates and slots for event schema induction. The main contribution of this paper are as follows:

- To better model the inner connectivity between entities, we borrow the normalized cut in image segmentation as the clustering criteria.
- We use constraints between templates and between slots in one sentence to improve *AESI* result.

Proceedings of NAACL-HLT 2016, pages 428–434,
San Diego, California, June 12-17, 2016.

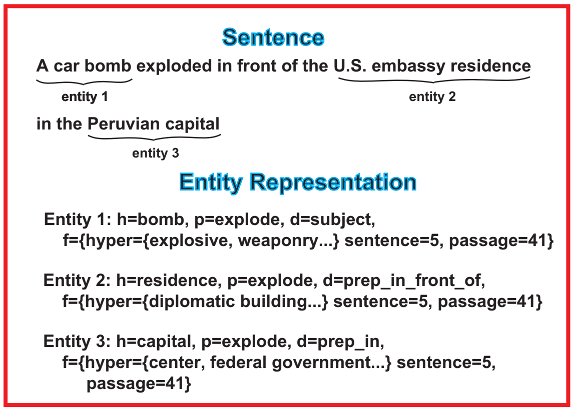

Figure 1: An entity example

2 Task Definition

Our model is an entity-driven model. This model represents a document d as a series of entities $E_d = \{e_i | i = 1, 2, \cdots\}$. Each entity is a quadruple $e = (h, p, d, f)$. Here, h represents the head word of an entity, p represents its predicate, and d represents the dependency path between the predicate and the head word, f contains the features of the entity (such as the **direct** hypernyms of the head word), the sentence id where e occurred and the document id where e occurred. A simple example is Fig 1.

Our ultimate goal is to assign two labels, a slot variable s and a template variable t, to each entity. After that, we can summarize all of them to get event schemas.

3 Automatic Event Schema Induction

3.1 Inner Connectivity Between Entities

We focus on two types of inner connectivity: (1) the likelihood of two entities to belong to the same template; (2) the likelihood of two entities to belong to the same slot;

3.1.1 Template Level Connectivity

It is easy to understand that entities occurred near each other are more likely to belong to the same template. Therefore, (Chambers and Jurafsky, 2011) uses PMI to measure the correlation of two words in the same document, but it cannot put two words from different documents together. In the Bayesian model of (Chambers, 2013), p(predicate) is the key factor to decide the template, but it ignores the fact that entities occurring nearby should belong to the same template. In this paper, we try to put two measures together. That is, if two entities occurred nearby, they can belong to the same template; if they have similar meaning, they can also belong to the same template. We use PMI to measure the distance similarity and use word vector (Mikolov et al., 2013) to calculate the semantic similarity.

A word vector can well represent the meaning of a word. So we concatenate the word vector of the j-th entity's head word and its predicate, denoted as $vec_{hp}(i)$. We use the cosine distance $\cos_{hp}(i, j)$ to measure the difference of two vectors.

Then we can get the template level connectivity formula as shown in Eq 1. The $PMI(i, j)$ is calculated by the head words of entity mention i and j.

$$W_T(i, j) = PMI(i, j) + \cos_{hp}(i, j) \quad (1)$$

3.1.2 Slot Level Connectivity

If two entities can play similar role in an event, they are likely to fill the same slot. We know that if two entities can play similar role, their head words may have the same hypernyms. We only consider the **direct** hypernyms here. Also, their predicates may have similar meaning and the entities may have the same dependency path to their predicate. Therefore, we give the factors equal weights and add them together to get the slot level similarity.

$$W_S(i, j) = \cos_p(i, j) + \delta(depend_i = depend_j) + \delta(hypernym_i \cap hypernym_j \neq \phi) \quad (2)$$

Here, the $\delta(\cdot)$ has value 1 when the inner expression is true and 0 otherwise. The "hypernym" is derived from Wordnet(Miller, 1995), so it is a set of direct hypernyms. If two entities' head words have at least one common direct hypernym, then they may belong to the same slot. And again $\cos_p(i, j)$ represents the cosine distance between the predicates' word vector of entity i and entity j.

3.2 Template and Slot Clustering Using Normalized Cut

Normalized cut intend to maximize the intra-class similarity while minimize the inter class similarity, which deals well with the connectivity between entities.

We represent each entity as a point in a high-dimension space. The edge weight between two points is their template level similarity / slot level similarity. Then the larger the similarity value is, the more likely the two entities (point) belong to the same template / slot, which is also our basis intuition.

For simplicity, denote the entity set as $E = \{e_1, \cdots, e_{|E|}\}$, and the template set as T. We use the $|E| \times |T|$ partition matrix X_T to represent the template clustering result. Let $X_T = [X_{T_1}, \cdots, X_{T_{|T|}}]$, where X_{T_l} is a binary indicator for template $l(T_l)$.

$$X_T(i,l) = \begin{cases} 1 & e_i \in T_l \\ 0 & otherwise \end{cases} \tag{3}$$

Usually, we define the degree matrix D_T as: $D_T(i,i) = \sum_{j \in E} W_T(i,j), i = 1, \cdots, |E|$. Obviously, D_T is a diagonal matrix. It contains information about the weight sum of edges attached to each vertex. Then we have the template clustering optimization as shown in Eq 4 according to (Shi and Malik, 2000).

$$\begin{aligned} \max \quad & \varepsilon_1(X_T) = \frac{1}{|T|} \sum_{l=1}^{|T|} \frac{X_{T_l}^T W_T X_{T_l}}{X_{T_l}^T D_T X_{T_l}} \\ s.t. \quad & X_T \in \{0,1\}^{|E| \times |T|} \quad X_T \mathbf{1}_{|T|} = \mathbf{1}_{|E|} \end{aligned} \tag{4}$$

where $\mathbf{1}_{|E|}$ represents the $|E| \times 1$ vector of all 1's.

For the slot clustering, we have a similar optimization as shown in Eq 5.

$$\begin{aligned} \max \quad & \varepsilon_2(X_S) = \frac{1}{|S|} \sum_{l=1}^{|S|} \frac{X_{S_l}^T W_S X_{S_l}}{X_{S_l}^T D_S X_{S_l}} \\ s.t. \quad & X_S \in \{0,1\}^{|E| \times |S|} \quad X_S \mathbf{1}_{|S|} = \mathbf{1}_{|E|} \end{aligned} \tag{5}$$

where S represents the slot set, X_S is the slot clustering result with $X_S = [X_{S_1}, \cdots, X_{S_{|S|}}]$, where X_{S_l} is a binary indicator for slot $l(S_l)$.

$$X_S(i,l) = \begin{cases} 1 & e_i \in S_l \\ 0 & otherwise \end{cases} \tag{6}$$

3.3 Joint Model With Sentence Constraints

For event schema induction, we find an important property and we name it "Sentence constraint". The entities in one sentence often belong to one template but different slots.

The sentence constraint contains two types of constraint, "template constraint" and "slot constraint".

1. **Template constraint**: Entities in the same sentence are usually in the same template. Hence we should make the templates taken by a sentence as few as possible.

2. **Slot constraint**: Entities in the same sentence are usually in different slots. Hence we should make the slots taken by a sentence as many as possible.

Based on these consideration, we can add an extra item to the optimization object. Let $N_{sentence}$ be the number of sentences. Define $N_{sentence} \times |E|$ matrix J as the sentence constraint matrix, the entries of J is as following:

$$J(i,j) = \begin{cases} 1 & e_i \in Sentence_j \\ 0 & otherwise \end{cases} \tag{7}$$

Easy to show, the product $G_T = J^T X_T$ represents the relation between sentences and templates. In matrix G_T, the (i,j)-th entry represents how many entities in sentence i are belong to T_j.

Using G_T, we can construct our objective. To represent the two constraints, the best objective we have found is the trace value: $tr(G_T G_T^T)$. Each entry on the diagonal of matrix $G_T G_T^T$ is the square sum of all the entries in the corresponding line in G_T, and the larger the trace value is, the less templates the sentence would taken. Since $tr(G_T G_T^T)$ is the sum of the diagonal elements, we only need to maximize the value $tr(G_T G_T^T)$ to meet the template constraint. For the same reason, we need to minimize the value $tr(G_S G_S^T)$ to meet the slot constraint.

Generally, we have the following optimization objective:

$$\varepsilon_3(X_T, X_S) = \frac{tr\left(X_T^T J J^T X_T\right)}{tr\left(X_S^T J J^T X_S\right)} \tag{8}$$

The whole joint model is shown in Eq 9. The detailed derivation[1] is shown in the supplement file.

$$
\begin{aligned}
X_T, X_S = \underset{X_T, X_S}{\operatorname{argmax}}\, & \varepsilon_1(X_T) + \varepsilon_2(X_S) + \varepsilon_3(X_T, X_S) \\
s.t.\ X_T \in \{0,1\}^{|E|\times|T|} \quad & X_T \mathbf{1}_{|T|} = \mathbf{1}_{|E|} \\
X_S \in \{0,1\}^{|E|\times|S|} \quad & X_S \mathbf{1}_{|S|} = \mathbf{1}_{|E|}
\end{aligned}
\tag{9}
$$

4 Experiment

4.1 Dataset

In this paper, we use MUC-4(Sundheim, 1991) as our dataset, which is the same as previous works (Chambers and Jurafsky, 2011; Chambers, 2013). MUC-4 corpus contains 1300 documents in the training set, 200 in development set (TS1, TS2) and 200 in testing set (TS3, TS4) about Latin American news of terrorism events. We ran several times on the 1500 documents (training/dev set) and choose the best $|T|$ and $|S|$ as $|T| = 6$, $|S| = 4$. Then we report the performance of test set. For each document, it provides a series of hand-constructed event schemas, which are called gold schemas. With these gold schemas we can evaluate our results. The MUC-4 corpus contains six template types: **Attack, Kidnapping, Bombing, Arson, Robbery, and Forced Work Stoppage**, and for each template, there are 25 slots. Since most previous works do not evaluate their performance on all the 25 slots, they instead focus on 4 main slots like Table 1, we will also focus on these four slots. We use the Stanford CoreNLP toolkit to parse the MUC-4 corpus.

Bombing

Perpetrator	Victim	Target	Instrument
El salvador	The police chief	ministry	explosives
The guerrillas	Students	The embassy	car bomb
The drag mafia	The Peruvian embassy	The police station	dynamite
Drug traffickers	The diplomat	organization	incendiary bomb
The Atlacatl battalion	soldiers	bridge	vehicle bomb

Attack

Perpetrator	Victim	Target	Instrument
troops	driver	organization	rifles
criminals	soldiers	helicopter	weapons
combat	children	person	gun
murder	civilians	livestock ministray building	explosives
person	journalists	vehicles	machinegun

Figure 2: Part of the result

	Prec	Recall	F1
C&J (2011)	**0.48**	0.25	0.33
Cheung (2013)	0.32	0.37	0.34
Chambers (2013)	0.41	0.41	0.41
Nguyen et al. (2015)	0.36	0.54	0.43
Our Model-SC	0.38	0.68	0.49
Our Model	0.39	**0.70**	**0.50**

Table 2: Slot-only mapping comparison to state-of-the-art unsupervised systems, "-SC" means without sentence constraint

4.2 Performance

Fig 2 shows two examples of our learned schemas: Bombing and Attacking. The five words in each slot are the five randomly picked entities from the mapped slots. The templates and slots that were joint learned seem reasonable.

Induced schemas need to map to gold schemas before evaluation. Previous works used two methods of mapping. The first ignores the schema type, and simply finds the best performing slot for each gold template slot. For instance, a perpetrator of a bombing and a perpetrator of an attack are treated the same. We call this the **slot-only mapping** evaluation. The second approach is to map each template t to the best gold template g, and limit the slot mapping so that only the slots under t can map to slots under g. We call this the **strict template mapping** evaluation. The slot-only mapping can result in higher scores since it is not constrained to preserve schema structure in the mapping.

We compare our results with four works (Chambers and Jurafsky, 2011; Cheung, 2013; Chambers, 2013; Nguyen et al., 2015) as is shown in Table 2 and Table 3. Our model has outperformed all of the previous methods. The improvement of recall is due to the normalized cut criteria, which can better use the inner connectivity between entities. The sentence constraint improves the result one step further.

Note that after adding the sentence constraint, the slot-only performance has increased a little, but the strict *template mapping* performance has increased a lot as is shown in Table 3. This phenomenon can be explained by the following facts: We count the

[1] At https://github.com/shalei120/ESI_1_2 can the code be found.

	Prec	Recall	F1
Chambers (2013)	**0.42**	0.27	0.33
Our Model-SC	0.26	**0.55**	0.35
Our Model	0.33	0.50	**0.40**

Table 3: strict template mapping comparison to state-of-the-art unsupervised systems, "-SC" means without sentence constraint

amount of entities which has been assigned different templates or different slots in "Our Model-SC" and "Our Model". Of all the 11465 entities, 2305 entities has been assigned different templates in the two methods while only 108 entities has different slots. This fact illustrates that the sentence constraint can affect the assignment of templates much more than the slots. Therefore, the sentence constraint leads largely improvement to the strict mapping performance and very little increase to the slot-only performance.

5 Related Works

The traditional information extraction task is to fill the event schema slots. Many slot filling algorithms requires the full information of the event schemas and the labeled corpus. Among them, there are rule-based method (Rau et al., 1992; Chinchor et al., 1993), supervised learning method (Baker et al., 1998; Chieu et al., 2003; Bunescu and Mooney, 2004; Patwardhan and Riloff, 2009; Maslennikov and Chua, 2007), bootstrapping method (Yangarber et al., 2000) and cross-document inference method (Ji and Grishman, 2008). Also there are many semi-supervised solutions, which begin with unlabeled, but clustered event-specific documents, and extract common word patterns as extractors (Riloff and Schmelzenbach, 1998; Sudo et al., 2003; Riloff et al., 2005; Patwardhan and Riloff, 2007; Filatova et al., 2006; Surdeanu et al., 2006)

Other traditional information extraction task learns binary relations and atomic facts. Models can learn relations like "Jenny is married to Bob" with unlabeled data (Banko et al., 2007; Etzioni et al., 2008; Yates et al., 2007; Fader et al., 2011), or ontology induction (dog is an animal) and attribute extraction (dogs have tails) (Carlson et al., 2010a; Carlson et al., 2010b; Huang and Riloff, 2010; Van Durme and Pasca, 2008), or rely on predefined patterns (Hearst, 1992).

Shinyama and Sekine (2006) proposed an approach to learn templates with unlabeled corpus. They use *unrestricted relation discovery* to discover relations in unlabeled corpus as well as extract their fillers. Their constraints are that they need redundant documents and their relations are binary over repeated named entities. (Chen et al., 2011) also extract binary relations using generative model.

Kasch and Oates (2010), Chambers and Jurafsky (2008), Chambers and Jurafsky (2009), Balasubramanian et al. (2013) captures template-like knowledge from unlabeled text by large-scale learning of scripts and narrative schemas. However, their structures are limited to frequent topics in a large corpus. Chambers and Jurafsky (2011) uses their idea, and their goal is to characterize a specific domain with limited data using a three-stage clustering algorithm.

Also, there are some state-of-the-art works using probabilistic graphic model (Chambers, 2013; Cheung, 2013; Nguyen et al., 2015). They use the Gibbs sampling and get good results.

6 Conclusion

This paper presented a joint entity-driven model to induct event schemas automatically.

This model uses word embedding as well as PMI to measure the inner connection of entities and uses normalized cut for more accurate clustering. Finally, our model uses sentence constraint to extract templates and slots simultaneously. The experiment has proved the effectiveness of our model.

Acknowledgments

This research is supported by National Key Basic Research Program of China (No.2014CB340504) and National Natural Science Foundation of China (No.61375074,61273318). The contact authors of this paper are Sujian Li and Baobao Chang.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume*

1, pages 86–90. Association for Computational Linguistics.

Niranjan Balasubramanian, Stephen Soderland, and Oren Etzioni Mausam. 2013. Generating coherent event schemas at scale. *Proceedings of the Empirical Methods in Natural Language Processing. ACM*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2670–2676. Morgan Kaufmann Publishers Inc.

Razvan Bunescu and Raymond J Mooney. 2004. Collective information extraction with relational markov networks. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 438. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010a. Toward an architecture for never-ending language learning. In *AAAI*.

Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. 2010b. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 101–110. ACM.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL*, pages 789–797.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 602–610. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. pages 976–986.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. EMNLP.

Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. 2011. In-domain relation discovery with meta-constraints via posterior regularization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 530–540. Association for Computational Linguistics.

Jackie Chi Kit Cheung. 2013. Probabilistic frame induction. *arXiv preprint arXiv:1302.4813*.

Hai Leong Chieu, Hwee Tou Ng, and Yoong Keok Lee. 2003. Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 216–223. Association for Computational Linguistics.

Nancy Chinchor, David D Lewis, and Lynette Hirschman. 1993. Evaluating message understanding systems: an analysis of the third message understanding conference (muc-3). *Computational linguistics*, 19(3):409–449.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 207–214. Association for Computational Linguistics.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2010. Inducing domain-specific semantic class taggers from (almost) nothing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 275–285. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*, pages 254–262.

Niels Kasch and Tim Oates. 2010. Mining script-like structures from the web. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 34–42. Association for Computational Linguistics.

Mstislav Maslennikov and Tat-Seng Chua. 2007. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the Association of Computational Linguistics (ACL)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 188–197, Beijing, China, July. Association for Computational Linguistics.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *EMNLP-CoNLL*, volume 7, pages 717–727.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 151–160. Association for Computational Linguistics.

Lisa Rau, George Krupka, Paul Jacobs, Ira Sider, and Lois Childs. 1992. Ge nltoolset: Muc-4 test results and analysis. In *Proceedings of the 4th conference on Message understanding*, pages 94–99. Association for Computational Linguistics.

Ellen Riloff and Mark Schmelzenbach. 1998. An empirical approach to conceptual case frame acquisition. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 49–56.

Ellen Riloff, Janyce Wiebe, and William Phillips. 2005. Exploiting subjectivity classification to improve information extraction. In *Proceedings of the National Conference On Artificial Intelligence*, volume 20, page 1106. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

R Sagayam, S Srinivasan, and S Roshni. 2012. A survey of text mining: Retrieval, extraction and indexing techniques. *International Journal Of Computational Engineering Research*, 2(5).

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 731–738. Association for Computational Linguistics.

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 224–231. Association for Computational Linguistics.

Beth Sundheim. 1991. Third message understanding evaluation and conference (muc-3): Phase 1 status report. In *HLT*.

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. A hybrid approach for the acquisition of information extraction patterns. *Adaptive Text Extraction and Mining (ATEM 2006)*, page 48.

Benjamin Van Durme and Marius Pasca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *AAAI*, volume 8, pages 1243–1248.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 940–946. Association for Computational Linguistics.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics.