Extracting Action Knowledge in Security Informatics

Ansheng Ge[1], Wenji Mao[1], Daniel Zeng[1], Qingchao Kong[1], Huachi Zhu[2]
[1]State Key Laboratory of Management and Control of Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]Beijing Police Department
{ansheng.ge, wenji.mao, dajun.zeng, qingchao.kong}@ia.ac.cn

***Abstract*—Actions are the primary way an entity interacts with other entities and acts on the external world. Action knowledge is of vital importance for behavior modeling, analysis and prediction in security informatics. In this paper, we present our approach to action knowledge extraction from Web textual data. Our approach is based on mutual bootstrapping with knowledge reasoning, which can acquire more action knowledge types and require less human participation compared with the related work. We evaluate the performance of our method and demonstrate its effectiveness through experiment.**

Keywords-action knowledge; bootstrapping; knowledge reasoning

I. INTRODUCTION

With the continuous development of Internet and the World Wide Web, extracting useful information from structured and unstructured textual data is becoming increasingly important. Most extraction works defined by MUC and ACE have concentrated on the recognition of entities, relations, events, and so on. However, an important class of knowledge extraction task, action knowledge extraction is almost totally ignored by the research community. On the other hand, actions are the primary way an entity interacts with other entities and acts on the external world. Thus action knowledge is of vital importance for behavior modeling, analysis and prediction in security informatics.

In recent years, there have been several studies concerning extracting common knowledge including action-related knowledge [1, 2]. Although these works have made good contributions to this field of research, the extracted action knowledge is often limited to certain types and the approaches usually require large amount of human efforts.

In this paper, we focus on three main types of action knowledge: action precondition, action effect and temporal relation. Take an action "*buy bomb*" as an example. Having *money* is the precondition of *buying bomb*, and having *bomb* is one of its effects. In addition, if another action "*get money*" leads to the effect of having *money*, it has a temporal relation with the action "*buy bomb*".

To ease human labor and promote the performance incrementally, we propose a semi-supervised learning method to action knowledge extract. In addition, we combine the learning method with knowledge reasoning approach to facilitate mutual leaning of action precondition and temporal relation. We conduct experimental study to compare our proposed approach with the alternative methods and verify the effectiveness of our approach.

The rest of our paper is organized as follows. In Section 2 we review the related work. Section 3 describes the system structure and workflow. Section 4 presents our action knowledge extraction approach in detail. Section 5 reports the results of our experimental study and we conclude our work in Section 6.

II. RELATED WORK

There have been knowledge bases related to actions such as VerbNet built by Kipper-Schuler[3]. Frames in VerbNet contain domain-independent action knowledge which includes syntactic description and semantic predicates. However, VerbNet only contains a limited set of actions, which is insufficient in security informatics. In addition, there is no knowledge about action relations and almost no knowledge about action preconditions in VerbNet.

Extracting causal relations has been studied previously. Khoo et al [4] used manually constructed linguistic templates to extract causal information from medical domain texts. Girju [5] presented an inductive learning approach to the automatic detection of lexical and semantic constraints for extracting causal relations between two noun phrases. Chang and Choi [6] employ bootstrapping to improve the performance of causal relation extraction. Extracting action knowledge is different from these previous works in two aspects. First, our focus is to find the causal relations between actions and states. Second, we need to extract richer knowledge types – not only causal relations, but also action preconditions.

Recently, Sil et al [1] proposed their action knowledge extraction method based on supervised learning. However, as their approach treated all the actions as verbs and they only tested 40 verbs from FrameNet, the performance of their approach in more complex domain is unclear. Li et al [2] treated an action as "verb+object", and used manually designed linguistic patterns to extract action preconditions and effects. Li's work is most similar to this work presented in the paper, but their approach is based on hand-made rules, which is laborious and time-consuming.

In addition, none of the above work has considered extracting temporal relations between actions, which is an important component of action knowledge and interrelated with both precondition and effect. To overcome the limitations in the related research, we proposed a statistical learning-based approach to action knowledge extraction. We take both action-state relations and temporal relations into consideration, extract different types of action knowledge, including action preconditions, effects and temporal relations, and compare the performance of our approach with alternative methods.

Bootstrapping has drawn much attention in information and knowledge extraction in recent years [6, 7]. It is a semi-supervised method that only needs a small number of seeds or templates. Without annotating any corpus data, it saves huge amount of human labor comparing with the rule-based approach. Besides, it can also incrementally improve the performance. Thus we use bootstrapping to extract action knowledge in our work. However, traditional bootstrapping is sensitive and unstable especially when seeds and templates are sparse in corpus. We compensate this with knowledge reasoning method.

III. System Structure and Workflow

The system structure and workflow are shown in Fig. 1. First, our system preprocesses the corpus and generates an action set [8]. The action set is then used to generate an effect set by referring to VerbNet. The system uses two bootstrapping processes to extract action preconditions and temporal relations, respectively. These two processes mutually facilitate each other with the help of knowledge reasoning. In the mutual bootstrapping process, with several manually constructed precondition templates, the system starts precondition bootstrapping first. During each iteration, after the system generates new precondition seeds, it tries to produce new temporal relation seeds with the help of reasoning. If there are new temporal relation seeds generated, similar bootstrapping process starts which generates new temporal relation seeds. The newly generated temporal relation seeds, in turn, are used to produce more precondition seeds with the help of reasoning. After sufficient many iterations, the most reliable templates and seeds are chosen as the output.

IV. Action Knowledge Extraction

Action knowledge extraction involves the mutual bootstrapping processes and the knowledge reasoning process. The mutual bootstrapping processes generate new action preconditions and temporal relations iteratively. The knowledge reasoning process helps produce action preconditions and temporal relations based on the interrelation of action knowledge. Action effects are generated using domain actions and the VerbNet.

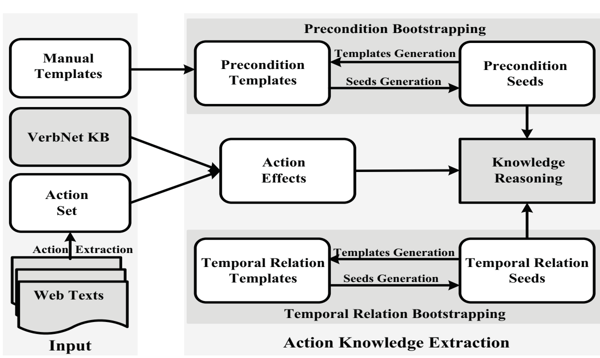

Figure 1. System Structure and Workflow

A. Effect Generation

VerbNet contains many semantic frames for each verb class. We make use of the frames "*has_possession(end(E), Agent, Theme)*"and "*exist(result(E), Product)*" to generate action effects. We represent an action as "verb+object". If the "verb" part of an action contains the semantic frames in their verb class in VerbNet, we then use the "object" part of the action as its effect.

B. Mutual Bootstrapping

At each iteration step, we calculate the reliability of the candidate seeds and templates. Then, the top-*k* reliability templates and seeds are chosen to run the next iteration, where *k* is the sum of the number of templates and seeds chosen in last iteration plus *n*. We set *n* to 2 and 5 for templates and seeds respectively. Similar to what has been proposed by Pantel and Pennacchiotti [9], we calculate the reliability of a template as follows:

$$r(t) = \frac{1}{|S_k| \times \max r(t)} \sum_{s \in S_k} pmi(s,t) \qquad (t \in T_k)$$

Where S_k and T_k are seed set and template set respectively, $pmi(s, t)$ is pointwise mutual information between the seed s and template t, and max $r(t)$ is the maximum reliability of templates in this iteration. We assign invariant value 1 to the reliability of each manual precondition template.

The reliability of a seed is calculated as follows:

$$r(s) = \frac{1}{|T_k| \times \max r(s)} \sum_{t \in T_k} pmi(s,t) \times r(t) \quad (s \in S_k)$$

Moreover, as the temporal relation is asymmetrical, we check the consistency of the generated temporal relations. For actions a_1 and a_2, if there exist both $<a_1, a_2>$ and $<a_2, a_1>$ in the temporal relation seeds, we discard both of them in this case.

C. Knowledge Reasoning

The types of action knowledge, action precondition, effect and temporal relation, are interrelated. For example, we can infer temporal relation from the knowledge of precondition and effect, and acquire precondition from the other two knowledge types. Below we describe their relations (Let a_1 and a_2 be actions and s_1 be a state):

$Effect(a_1,s_1) \wedge Precondition(a_2,s_1) \Rightarrow Temporal\text{-}relation(a_1,a_2)$

$Effect(a_1,s_1) \wedge Temporal\text{-}relation(a_1,a_2) \Rightarrow Precondition(a_2,s_1)$

After the generation of precondition seeds and temporal relation seeds at the end of each iteration, we will mutually expand each seed set by reasoning as follows: for each precondition seed $<a_2, s_1>$, we check whether s_1 is in the effect set; if so, we add the associated action a_1 of effect s_1 and a_2 ($<a_1, a_2>$) to temporal relation seed set. Similarly, we also check each $<a_1, a_2>$ in the temporal relation seed set and see whether a_1 is associated to a state s_1 in the effect set; if so, we add seed $<a_2, s_1>$ to the precondition seed set. However, if the reasoning results are already in the seed sets, the system will ignore them.

V. EXPERIMENTS

We use Web corpus in security domain to test our method. The textual data are the news about Al-Qaeda reported in The Times, BBC, USA TODAY, The New York Times and The Guardian, with totally 25,103 pages. After preprocessing using Stanford NLP tool[1], we collect 851,689 sentences in total. We then extract and filter actions from all these sentences using chunk tool[2], with each action in the form “verb+object”. At last, we get 21,259 actions with quality.

To start action knowledge extraction, we only use two manually designed precondition templates, “use <state> to <action>” and “need <state> to <action>”. The system generates the effect set based on the VerbNet, and recursively employ mutual bootstrapping and knowledge reasoning to generate action preconditions and temporal relations. To test the effectiveness of our method, we compare our method with two alternative methods. We choose the following two methods as the alternative methods:

1) Rule-based Method

To compare our method with the related work, we directly use the manually constructed precondition templates in [3] and assign reliability value 1 to all these templates. As the results after the first iteration indicate the performance of this method, we select the same number of preconditions generated by this method and ours for comparison.

2) Bootstrapping Method

To test the performance of our mutual bootstrapping method with reasoning, we use traditional bootstrapping without reasoning as the baseline. This method uses the same original precondition templates as our method and iterates 20 times. As there is no reasoning process involved, only precondition bootstrapping can execute within this method.

Table II shows the results of rule-based method, bootstrapping method and our method after 20 iterations. We can see from the table that the precision of preconditions of our method is 79%, which is 5% higher than that of the pure bootstrapping method and 16% higher than that of the rule-based method. In addition, the precision of temporal relations of our method is 72.6%. The experimental results generally verify the effectiveness of our method in extracting action knowledge.

In addition, we further explore the precision curves of preconditions in different methods. We can see from Fig. 2 that the slowdown of the precision in our method is slower than that of the pure bootstrapping method. This means the reliabilities of seeds in our method are more trustworthy. We can also see from the figure that the precision of rule-based method is clearly lower than those of the bootstrapping based methods.

TABLE I. EXPERIMENTS RESULTS OF DIFFERENT METHODS

Methods	Precondition precision	Temporal relation precision
Rule-based method	63%	/
Baseline method	74%	/
Bootstrapping with reasoning method	79%	72.6%

[1] http://nlp.stanford.edu/software/corenlp.shtml
[2] http://ilk.uvt.nl/team/sabine/chunklink/README.html

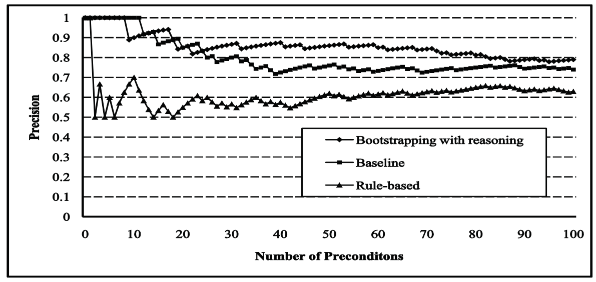

Figure 2. Precondition precision curves of top-100 results

VI. CONCLUSION

In this paper, we present our approach to extracting action knowledge from Web textual data. Based on mutual bootstrapping with knowledge reasoning, our method can acquire three types of action knowledge, action preconditions, effects and temporal relations. Compared with the related work, our method requires relatively less human labor and can incrementally improve the performance. The experimental results demonstrate the effectiveness of our method in security informatics.

ACKNOWLEDGMENT

This work is supported in part by the NNSFC Grant Nos. 60921061, 61175040, 71025001, 90924302 and 91024030, and the Chinese Academy of Sciences under Grant No 2F11N06.

REFERENCES

[1] A. Sil, *et al.*, "Extracting action and event semantics from web text," in *AAAI 2010 Fall Symposium on Commonsense Knowledge (CSK)*, 2010.
[2] X. Li, *et al.*, "Automatic construction of domain theory for attack planning," in *2010 IEEE International Conference on Intelligence and Security Informatics (ISI 2010)*, 2010, pp. 65-70.
[3] K. Kipper-Schuler, "VerbNet: A broad-coverage, comprehensive verb lexicon," *Ph.D. thesis*, Computer and Information Science Dept., University of Pennsylvania, Philadelphia, PA, 2005.
[4] C. S. G. Khoo, *et al.*, "Extracting causal knowledge from a medical database using graphical patterns," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000, pp. 336-343.
[5] R. Girju, "Automatic detection of causal relations for question answering," in *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, 2003.
[6] D.-S. Chang and K.-S. Choi, "Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities," *Information Processing and Management,* vol. 42, pp. 662-678, 2006.
[7] S. Liao and R. Grishman, "Filtered Ranking for Bootstrapping in Event Extraction," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.
[8] A. Ge, W. Mao, D. Zeng and Q. Kong, "Extracting Action Knowledge from Texts," in *Proceedings of the Third Chinese Conference on Social Computing*, 2011.
[9] P. Pantel and M. Pennacchiotti, "Espresso: Leveraging generic patterns for automatically harvesting semantic relations," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006, pp. 113-120.