Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn



BUEES: a bottom-up event extraction system*#

Xiao DING, Bing QIN, Ting LIU‡

(*Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin 150001, China*)

E-mail: xding@ir.hit.edu.cn; bqin@ir.hit.edu.cn; tliu@ir.hit.edu.cn

Received Nov. 27, 2014; Revision accepted May 11, 2015; Crosschecked June 8, 2015

Abstract: Traditional event extraction systems focus mainly on event type identification and event participant extraction based on pre-specified event type paradigms and manually annotated corpora. However, different domains have different event type paradigms. When transferring to a new domain, we have to build a new event type paradigm and annotate a new corpus from scratch. This kind of conventional event extraction system requires massive human effort, and hence prevents event extraction from being widely applicable. In this paper, we present BUEES, a bottom-up event extraction system, which extracts events from the web in a completely unsupervised way. The system automatically builds an event type paradigm in the input corpus, and then proceeds to extract a large number of instance patterns of these events. Subsequently, the system extracts event arguments according to these patterns. By conducting a series of experiments, we demonstrate the good performance of BUEES and compare it to a state-of-the-art Chinese event extraction system, i.e., a supervised event extraction system. Experimental results show that BUEES performs comparably (5% higher F-measure in event type identification and 3% higher F-measure in event argument extraction), but without any human effort.

Key words: Event extraction, Unsupervised learning, Bottom-up
doi:10.1631/FITEE.1400405 **Document code:** A **CLC number:** TP391

1 Introduction

Information extraction (IE) is a task of identifying factual description (entities, relations, and events) from unstructured natural language text and extracting information related to those descriptions (Grishman, 1997). Event extraction remains the most challenging task, because a large field of view is often needed to understand how facts tie together, and it is situated at the end of an IE pipeline and thus suffers from the propagation of errors from word segmentation, named entity recognition, coreference resolution, etc. (Ahn, 2006). Although event extraction is a challenging problem, it has been widely used in several different specific domains, such as musical reports (Ding *et al.*, 2011), financial analysis (Lee *et al.*, 2003), biomedical investigation (Pham *et al.*, 2013), and legal documents (Schilder, 2007).

The main approaches used by most event extraction systems are based on knowledge engineering technology or machine learning technology. The knowledge engineering based event extraction systems use extraction patterns or rules to identify and extract the relevant information (Riloff, 1996; Soderland, 1999; Yangarber *et al.*, 2000). Most of these systems use annotated training data to learn pattern matching rules, based on lexical, syntactic, or semantic information. These systems traditionally were the top performers in most event extraction benchmarks, such as Message Understanding Conference (MUC) (Chinchor *et al.*, 1993) and automatic content extraction (ACE) (Yeh *et al.*, 2002). In the machine learning approach, domain experts label

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61133012 and 61472107) and the National Basic Research Program (973) of China (No. 2014CB340503)

A preliminary version was presented at the 6th International Joint Conference on Natural Language Processing, Oct. 14-18, 2013, Japan

ORCID: Xiao DING, http://orcid.org/0000-0002-5838-0320

©Zhejiang University and Springer-Verlag Berlin Heidelberg 2015

instances of the target concepts in a set of documents (Miwa *et al.*, 2010; Hong *et al.*, 2011; Ritter *et al.*, 2012). The system then learns a model of the extraction task, which can be applied to new documents automatically.

Both of these approaches require substantial human effort, and hence prevent event extraction systems from being domain adaptive and more widely applicable. Recently, many semi-supervised event extraction systems have been proposed that aim to reduce the annotated data required, ideally to a set of seed instances of the target events. One such system was introduced by Liao and Grishman (2010). They used two state-of-the-art bootstrapping-based event extraction systems, and then ranked their candidate patterns and accepted the top-ranked patterns in each iteration.

However, for all such approaches it is still necessary to specify the target events in advance. In this paper, we explore the possibility of constructing a completely unsupervised and bottom-up event extraction system, which does not need to pre-specify the target event types.

The task is important, and the challenge of it lies at least in two aspects:

1. How to automatically build an event type paradigm. As pre-specifying interested event types in a domain needs rich background knowledge, an event type paradigm is traditionally built by domain experts. It is costly work.

2. How to extract event arguments in a totally unsupervised way. All of the event extraction systems reported in the literature need manual efforts to a greater or lesser extent.

To address the above challenges, in this study, we have designed and developed a bootstrapping-based bottom-up event extraction system, BUEES. The system automatically builds an event type paradigm from scratch, based on the definition of an event trigger, "the words that most clearly express an event's occurrence", and our key observations "Triggers are the most important lexical units to represent events. A set of triggers with similar meaning or usage represents the same event type". Event types can be discovered based on trigger clustering.

When the target event types are available, the next step is to learn a set of patterns to extract event arguments from web documents. The system takes as input a small set of event seeds. It then uses these seeds to search the web to obtain more documents that contain the event seeds. The extraction patterns are learned from these documents. Finally, we can extract event arguments by using these useful patterns.

The major contributions of the work presented in this paper are as follows:

1. To the best of our knowledge, our work is the first to propose the bottom-up event extraction system. We automatically build an event type paradigm. Based on the paradigm, we proceed to implement traditional event extraction tasks.

2. Our system is completely unsupervised. As far as we know, all of the bootstrapping-based semi-supervised event extraction systems need the manually constructed seeds in advance. In contrast, our system can generate and select event seeds automatically.

2 Task description

2.1 ACE event extraction task

The event extraction task we are addressing is the ACE evaluations (http://projects.ldc.upenn.edu/ace/docs/English-Events-Guidelines_v5.4.3.pdf), where an event is defined as a specific occurrence involving participants. An event extraction task requires that certain specified types of events should be detected. We first introduce ACE terminology:

1. Entity: an object or a set of objects in one of the semantic categories of interest.

2. Entity mention: a reference to an entity (typically, a noun phrase).

3. Event trigger: the main word that most clearly expresses an event occurrence.

4. Event arguments: the entity mentions that are involved in an event.

5. Event mention: a phrase or sentence within which an event is described, including trigger and arguments.

6. Event type: a particular event category, such as 'Conflict/Attack' and 'Life/Die'.

The ACE 2005 evaluation has 8 types of events, with 33 subtypes. We treat these simply as 33 separate event types and do not consider the hierarchical structure among them. In addition, the ACE evaluation plan defines the following standards to determine the correctness of an event extraction:

1. A trigger is correctly labeled if its event type and offset (i.e., the position of the trigger word in text) match a reference trigger.

2. An argument is correctly identified if its event type and offsets match any of the reference argument mentions; in other words, participants are correctly recognized in an event.

3. An argument is correctly classified if its role matches any of the reference argument mentions.

Fig. 1 shows an example of an ACE event, where 'born' is the trigger word. Its event type is 'Life' and the subtype is 'Be-born'. This event consists of three arguments, namely 'Mao Ze-dong', '1893', and 'Xiangtan, Hunan Province', which correspond to three role labels in the Life/Be-born event template of 'Person', 'Time-within', and 'Place', respectively.

Note that we use the 863 POS tagging label set (http://www.ltp-cloud.com/intro/#pos) throughout the paper.

2.2 New task for event extraction

In addition to traditional event extraction tasks introduced above, we propose a new task of building an event type paradigm. ACE manually annotates 8 types and 33 subtypes of events and constructs the event type paradigm as shown in Table 1. However, building an event type paradigm in this way not only requires massive human effort but also tends to be data dependent. As a result, it may prevent the event extraction from being widely applicable. Since event types among domains are different, the event type paradigm of ACE, which does not define, for example, music-related events, is useless for music domain event extraction. So, we would have to build a totally different event type paradigm for the music domain from scratch.

Event: Mao Ze-dong was born in Xiangtan, Hunan Province in 1893.
Event type: Life
Event subtype: Be-born
Trigger: born
Arguments:
- **Person:** Mao Ze-dong
- **Time-within:** 1893
- **Place:** Xiangtan, Hunan Province

Fig. 1 An example of event extraction

Table 1 ACE event type paradigm

Type	Subtype
Life	Be-born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-ownership, Transfer-money
Business	Start-org, Merge-org, Declare-bankruptcy, End-org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-write
Personnel	Start-position, End-position, Nominate, Elect
Justice	Arrest-jail, Release-parole, Trial-hearing, Charge-indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

3 Description of BUEES

The goal of BUEES is to extract instances of events without any human supervision. The system is based on the framework of bootstrapping and its architecture is as shown in Fig. 2. Traditional bootstrapping-based semi-supervised event extraction systems use manual construction of seed examples to learn extraction patterns, and then identify event types and recognize event arguments. Since the number of seeds is limited, the quality and coverage of seeds highly affect the performance of extraction patterns. However, we propose to automatically construct the set of seeds (Sections 3.1–3.3) and explore a novel extraction pattern learning algorithm (Sections 3.4 and 3.5).

The system works in three stages. During the first stage, the system builds the event type paradigm and prepares seed instances. During the second stage, the system learns extraction patterns based on the seed set. During the third stage, the system identifies event types and extracts event arguments based on the learned patterns. In the following sections, we introduce each component of BUEES in detail.

3.1 Event type building and seed extractor

Since the event trigger is the word that most clearly expresses an event's occurrence, the key idea

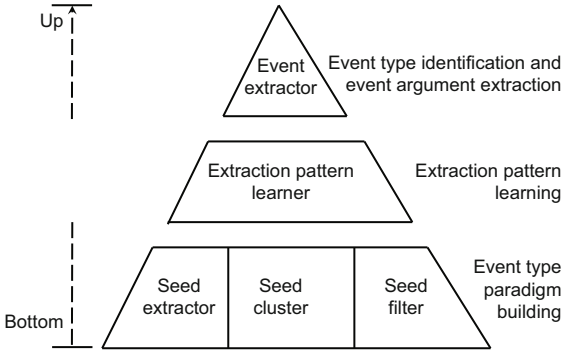


Fig. 2 The architecture of BUEES

of this paper is to automatically construct an event type paradigm by clustering event triggers. For example, in the ACE corpus, a set of event triggers {倒闭、闭门、关闭、停业、解散} ({bankrupt, shut down, close, close down, dismiss}) represents the sense of the event type 'Business/End-org'. In addition to extracting the event trigger in the sentence, we extract its subject and object as the seed instance (subject, trigger, object). The event seeds are used in two ways. First, the event type paradigm is built by clustering seeds. Second, the seeds are used for learning extraction patterns (see Section 3.4).

Sudo *et al.* (2003) summarized three classical models for representing events. All of these three models rely on the syntactic tree structure, and the trigger is specified as a predicate in this structure. To accurately extract event seeds, we employ the predicate-argument model (Yangarber *et al.*, 2000; Ding *et al.*, 2013), which is based on a direct syntactic relation between a predicate and its arguments. We extract the syntactic relation for the predicate-argument model by means of the HIT (Harbin Institute of Technology) Dependency Parser (Che *et al.*, 2009). Based on the predicate-argument model, we propose a seed extraction (SE) algorithm. The details are shown in Algorithm 1.

Algorithm 1 Seed extraction

Require: Raw corpus D
Ensure: Candidate seeds
for document d in raw corpus D **do**
  $d \leftarrow$ Paragraph splitting
  $d \leftarrow$ Sentence splitting
  for sentence s in document d **do**
    $s \leftarrow$ Word segmentation
    $s \leftarrow$ Chinese dependency parsing
    $s \leftarrow$ Identify subject-predicate relation (SBV) pair (V_{SBV}, Sub) and verb-object relation (VOB) pair (V_{VOB}, Obj)
    if $V_{\mathrm{SBV}} = V_{\mathrm{VOB}} = V_t$ **then**
      Extract V_t as a candidate trigger
      Extract $(\mathrm{Sub}, V_t, \mathrm{Obj})$ as a candidate seed
    end if
  end for
end for

Take the following sentence as an example:

$$\underset{1}{\text{毛泽东}}\ \underset{2}{\text{1893年}}\ \underset{3}{\text{出生}}\ \underset{4}{\text{于}}\ \underset{5}{\text{湖南湘潭}} \rightarrow$$
$$\underset{1}{\text{Mao Ze-dong}}\ \underset{2}{\text{was}}\ \underset{3}{\text{born}}\ \underset{4}{\text{in}}\ \underset{5}{\text{Xiangtan, Hunan Province}}$$

$$\underset{6}{\text{in}}\ \underset{7}{\text{1893}}$$

The HIT Chinese Dependency Parser dependencies are:

SBV (出生-3, 毛泽东-1)
  → (born-3, Mao Ze-dong-1)
VOB (出生-3, 湖南湘潭-5)
  → (born-3, Xiangtan, Hunan Province-5)
ADV (出生-3, 1893年-2)
  → (born-3, 1893-7)
POB (湖南湘潭-5, 于-4)
  → (Xiangtan, Hunan Province-5, in-4)

where each individual formula represents a binary dependence from the governor (the first token) to the dependent (the second token). The SBV relation, which stands for the subject-predicate structure, means that the head is a predicate verb and the dependent is a subject of the predicate verb; the VOB dependency relation, which stands for the verb-object structure, means that the head is a verb and the dependent is an object of the verb; the ADV relation, which stands for the adverbial structure, means that the head is a verb and the dependent is an adverb of the verb; the POB relation, which stands for the prep-object structure, means that the head is an object and the dependent is a preposition of the object.

Since $V_{\mathrm{SBV}} = V_{\mathrm{VOB}} = V_t =$ 出生 (born) in this case, based on the predicate-argument model, the word '出生(born)' should be extracted as a candidate event trigger and (Mao Ze-dong, born, Xiangtan, Hunan Province) should be extracted as a candidate event seed instance.

3.2 Seed cluster

As discussed above, a set of triggers with the same meaning and usage represents the same event type. We propose to cluster event seeds (i.e., event trigger and its corresponding subject and object) based on their semantic distances, and each of these clusters represents a type of event. Details are shown in Algorithm 2.

For every two seeds p_i and p_j $(i \neq j)$ in Algorithm 2, the similarity function $\mathrm{Sim}(p_i, p_j)$ is calculated using semantic information provided by HowNet (Dong and Dong, 2006) as

$$\mathrm{Sim}(p_i, p_j) = \frac{2 \cdot \mathrm{Sum}_s}{\mathrm{Sum}_i + \mathrm{Sum}_j}, \tag{1}$$

with

$$\begin{cases} \mathrm{Sum}_s = N_s + S_s + O_s, \\ \mathrm{Sum}_i = N_i + S_i + O_i, \\ \mathrm{Sum}_j = N_j + S_j + O_j, \end{cases} \quad (2)$$

where S_s and O_s denote the numbers of identical sememes in DEF_s (the concept definition in HowNet) of Sub_i and Sub_j, Obj_i and Obj_j respectively, S_i and S_j denote the numbers of sememes in DEF_s of Sub_i and Sub_j, respectively, and O_i and O_j denote the numbers of sememes in DEF_s of Obj_i and Obj_j, respectively. HowNet uses sememes to interpret concepts. Sememes are regarded as the basic unit of the meaning. For example, 'paper' can be viewed as a concept, and its sememes are 'white', 'thin', 'soft', 'flammable', etc.

A group of event seeds are aggregated to a seed cluster according to their semantic distance, and we view each seed cluster as one kind of event type. Then all these event types are employed to construct an event type paradigm.

Algorithm 2 Seed cluster (SC)

Require: Candidate seeds $(\mathrm{Sub}, V_t, \mathrm{Obj})$, threshold θ
Ensure: Event clusters EC
1: $\mathrm{EC} \leftarrow \varnothing$
2: **for** seed p in the set of seeds P **do**
3: Compute the similarity (Sim) between p and the other seeds
4: **if** $\mathrm{Sim} \geq \theta$ **then**
5: Add V_t to the related event type $\mathrm{ET}_{\mathrm{re}} \cup \{V_t\}$
6: **else if** $\mathrm{Sim} < \theta$ **then**
7: Set up a new event type $\mathrm{ET}_{\mathrm{new}}$
8: $\mathrm{EC} \leftarrow \mathrm{ET}_{\mathrm{new}}$
9: **end if**
10: **end for**

3.3 Seed filter

Although we have obtained some useful candidate seeds, certain meaningless candidate seeds come along in the results of the seed extractor as well. Therefore, we introduce a seed filter which uses a heuristic rule and ranking algorithm to filter out these less informative antecedent candidates.

Since event trigger words are extracted based on the predicate-argument model, most of these candidate trigger words are verb terms. However, not all verb terms can be used as trigger words. For example, the copular verb (e.g., 'is') rarely acts as the event trigger. To investigate which categories of verbs can serve as event triggers, we classify Chinese verbs into eight subclasses (Table 2). Such classification makes each subclass function as one grammatical role. For example, a modal verb will never be the predicate of a sentence and a nominal verb will always function as a noun.

Table 2 The scheme of verb subclass

Verb	Description	Example
vx	Copular verb	他是对的 (He is right)
vz	Modal verb	你应该努力工作 (You should work hard)
vf	Formal verb	他要求予以澄清 (He'd demand an explanation)
vq	Directional verb	他认识到困难 (He has realized the difficulties)
vb	Resultative verb	他看完了电影 (He has seen the movie)
vg	**General verb**	他**喜欢**踢足球 (He likes playing football)
vn	**Nominal verb**	参加我们的**讨论** (Take part in our discussion)
vd	Adverbial verb	产量持续增长 (Production increases steadily)

Bold term indicates that this kind of verb is required

We perform the verb sub-classification model based on the work by Liu *et al.* (2007) (provided by the Research Center for Social Computing and Information Retrieval in Harbin Institute of Technology, China). Statistically, about 94% of ACE Chinese event triggers are general verbs or nominal verbs and other types of verbs are rarely trigger words. To ensure the accuracy of event seed instances, we stress that the trigger word in a candidate event seed must be a general or nominal verb.

3.4 Instance collector and pattern learner

The instance collector is currently implemented in a very simple way. It takes event seeds as input, and then a search engine is used to retrieve documents that contain at least one of the seed words.

Fig. 3 depicts a general procedure of pattern learning. First, the collected sentences are used to generate the event instances which are tagged with part of speech (POS) tagging and named entity (NE) tagging. Then the NE labels are replaced by '[SLOT]' marks.

For example, assuming there is a seed (Mao Ze-dong, born, Xiangtan, Hunan Province), we can

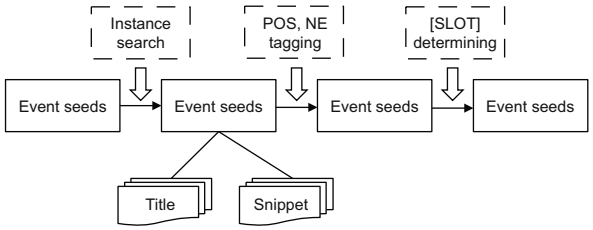


Fig. 3 The procedure of pattern learning

obtain the related sentence from the search engine as shown in Example 1.

Example 1 Mao Ze-dong was born in Xiangtan, Hunan Province in 1893.

The sentence will be represented as an event instance as shown in Example 2.

Example 2 Mao Ze-dong/Nh was/v born/v in/p Xiangtan, Hunan Province/Ns in/p 1893/Nr ./wp

The named entity is replaced by [SLOT] marks as shown in Example 3.

Example 3 [SLOT1]/Nh [SLOT2]/Nr born/v in/p [SLOT3]/Ns ./wp

The event instance patterns are generated as mentioned above.

3.5 Soft-pattern learner

To improve the generalization ability of the extraction pattern, we employ a soft-pattern of the sequential pattern as the final output of the pattern learner.

In our system, the soft-pattern consists of the following symbols:

1. Slots: matching event entities;

2. Tokens: including non-NE and POS tagging;

3. Skips (denoted by *): matching zero or more arbitrary tokens.

Soft-patterns are generalized from the set of event instance patterns. We exploit the soft-pattern learning algorithm (SPL) as shown in Algorithm 3.

The core algorithm in the generalization function is the best match algorithm (Friedman *et al.*, 1977), which is based on the longest common sequence (LCS) algorithm (Hirschberg, 1977). We modify the LCS algorithm with matching cost:

1. Two match units are identical: cost=0.

2. Two match units share the same NE type but different NE values: cost=5.

3. If both of two match units are noun, verb, adjective, or adverb, compare their labels in a thesaurus TongYiCi CiLin (expansion version) (Mei *et al.*, 1983) (the dictionary is recorded and expanded by the Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology). If there are some overlapping labels between two match units, the cost is 5; otherwise, cost=10.

4. No match of two units: cost=10.

Algorithm 3 Soft-pattern learning (SPL)

```
for event type T do
  for setence pair S_i, S_j from the Pattern Instance Set (T) do
    S_i is generated by seed (e_i1, e_i2)
    S_j is generated by seed (e_j1, e_j2)
    if entity type e_i1 = e_j1 and entity type e_i2 = e_j2 then
      Let Pattern=Generalization(S_i, S_j)
      Add Pattern to Soft-Pattern Set (T)
    end if
  end for
end for
```

After the best match is found, the event instance is converted into a soft-pattern by copying matched identical elements, adding skips and slots. For example, assume that we obtain the following two sentences by feeding the event seed into the search engine:

周杰伦本年度新专辑《魔杰座》将于10月9日全亚洲同步发行。

→ This year, Jay Chou's new album 'Mo Jie Zuo' will be released all over Asia on October 9.

蔡依林全新大碟《花蝴蝶》已于2009年3月27日全球同步发行。

→ Jolin Tsai's new album 'Butterfly' was released all over the world on March 27, 2009.

Event instances can be generated as follows by using the approach introduced in Section 3.4:

[SLOT1]/Nh 本/n 年 度/n 新/a 专 辑/n [SLOT2]/Nb 将/d 于/p [SLOT3]/Nr 全/a 亚洲/Ns 同步/v 发行/v 。/wp

[SLOT1]/Nh 全新/b 大碟/n [SLOT2]/Nb 已/d 于/p [SLOT3]/Nr 全球/n 同步/v 发行/v 。/wp

The best match can be found based on the LCS algorithm. Then the soft-pattern can be generated as follows:

* [SLOT1]/Nh * * Eb28A01=新/a Dk21B07=专辑/n [SLOT2]/Nb * 于/p [SLOT3]/Nr * * Jb01A10# Ka23A01 同步/v Hd13D02= He03B09= 发行/v * *

Note that the numbers ('Eb28A01') before '新', etc. are synonym labels from TongYiCi CiLin (expansion version). According to the costs defined

above, the soft-pattern learning algorithm is able to find the best generalization of any two event instance patterns. An example of the best match algorithm is shown in Table 3.

Table 3 An example of the best match algorithm

Pattern instance 1	Pattern instance 2	Cost
[SLOT1]/Nh	[SLOT1]/Nh	0
本/n (this)		10
年度/n (year)		10
新/a (new)	全新/b (new)	5
专辑/n (album)	大碟/n (album)	5
[SLOT2]/Nb	[SLOT2]/Nb	0
将/d (will be)		10
	已/d (was)	10
于/p (on)	于/p (on)	0
[SLOT3]/Nr	[SLOT3]/Nr	0
全/a (all)		10
	全球/n (world)	10
亚洲/Ns (Asia)		10
同步/v (simultaneous)	同步/v (simultaneous)	0
发行/v (release)	发行/v (release)	0
。/wp	。/wp	0
Total		80

4 Experiments

To evaluate the effectiveness of our BUEES, we first evaluate the performance of the proposed event type paradigm building approach. Then we compare our event extraction approach with a state-of-the-art baseline.

4.1 Data description

We use the ACE 2005 corpus, the same as that used in the baseline system, for our experiment. The corpus contains 633 Chinese documents, which are categorized by three genres: Newswire, Broadcast News, and Weblog. We randomly select 558 documents for event type paradigm building and 66 documents as a test set for event extraction. We use the ACE 2005 event type paradigm as the gold standard paradigm to evaluate our proposed approach.

To evaluate the robustness of our approach, we also use two specific domain data sets: Financial News (http://www.10jqka.com.cn/) and Musical News (http://yue.sina.com.cn/), collected by ourselves. The domain-specific corpus contains 6000 sentences from Financial News and 6000 sentences from Musical News. We carefully conduct user studies in two specific domain corpora. For each sentence in the data, two annotators are asked to label and cluster all potential triggers. The agreement between our two annotators, measured using Cohen's Kappa coefficient, is substantial (Kappa=0.75). We ask a third annotator to adjudicate the trigger clusters on which the former two annotators disagreed. Each trigger cluster is used to represent one type of event. All these events construct our final event type paradigm.

4.2 Event type paradigm building

We first propose the task of building the event type paradigm. To evaluate the effectiveness of our approach, we explore several reasonable evaluation metrics and implement a natural baseline method. The details are as follows.

4.2.1 Evaluation metrics

We adopt the F-measure (F) and Purity (Halkidi *et al.*, 2001) to determine the correctness of an event cluster:

$$p(i,r)=\frac{n(i,r)}{n_r},\ r(i,r)=\frac{n(i,r)}{n_i}, \tag{3}$$

$$f(i,r)=\frac{2\cdot p(i,r)\cdot r(i,r)}{p(i,r)+r(i,r)}, \tag{4}$$

$$F=\sum_i \frac{n_i}{n}\max\{f(i,r)\}, \tag{5}$$

$$\text{Purity}=\sum_r \frac{n_r}{n}\max\{p(i,r)\}, \tag{6}$$

where i is the gold standard event seed cluster, r is the event seed cluster that has the most identical seeds to i, n_i and n_r are the numbers of seeds in clusters i and r respectively, n is the total number of seeds, and $n(i,r)$ is the number of seeds identical between i and r. For every cluster we first compute $p(i,r)$, $r(i,r)$, and $f(i,r)$. Then we obtain F-measure and Purity for the whole clustering result. Note that the evaluation is based on word instances rather than word types.

4.2.2 Baseline method

Since this is the first bottom-up process, there is no directly comparable work. We build an event type paradigm based on clustering event seeds that consist of an event trigger and its corresponding subject and object. A natural baseline method for this problem

is only clustering event triggers. A group of triggers are aggregated to a trigger cluster according to their semantic distance, and we view each trigger cluster as one kind of event type. Then all these event types are employed to construct an event type paradigm.

4.2.3 Results and analysis

We first evaluate the task of event type paradigm building. All the evaluation results are shown in Table 4. As we have said, there is no previous work on this problem, and thus the comparison experiments are implemented on our two different approaches. The baseline method clusters only trigger words. In contrast, our approach clusters event seeds.

Table 4 Experimental results of event type paradigm building

Corpus	F-measure		Purity	
	Baseline	Our method	Baseline	Our method
ACE	63.21%	69.57%	60.17%	70.24%
FN	71.52%	74.42%	74.81%	76.18%
MN	63.21%	75.08%	68.17%	80.28%

FN: Financial News; MN: Musical News

Table 4 shows that the F-measure score is boosted from 63.21% to 69.57% and the Purity score from 60.17% to 70.24% using our approach. The reasons for this are as follows.

A trigger word itself is not enough for representing an event. The trigger and its corresponding subject and object play an important role in the event type discovery algorithm. Referring to in Section 3.1, most trigger words are verb terms. Polysemic verbs are a major issue in the natural language processing (NLP) community, such as 'to fire a gun' and 'to fire a manager', where 'fire' has two different meanings. The state-of-the-art verb sense disambiguation approach (Wagner *et al.*, 2009) stresses that verbs which agree on their selectional preferences belong to a common semantic class, for example, 'to arrest the suspect' and 'to capture the suspect'. Hence, our approach can achieve better performance than the baseline method.

We also run the comparison experiment using three different corpora (ACE 2005, Financial News, and Musical News) to evaluate the robustness and domain adaptiveness of our system. The performances on the specific domain corpora are better than that on the ACE corpus (about 5% absolute improvement on F-measure and 6%–10% on Purity). The main reason is that the events in the specific domain are more specific. In addition, results on both specific domain corpora achieve a good performance. This indicates that our system is domain independent.

4.2.4 Analysis of experimental errors

We first inspect the errors produced by our approach. The errors are caused mainly by the sparse event triggers in the corpus. Table 5 shows the distribution of the errors in detail.

Table 5 Experimental error distributions of event type paradigm building

Error type	Proportion	Error type	Proportion
Trigger extraction	33.0%	Trigger filter	19.5%
Trigger ambiguity	28.3%	Others	19.2%

After error analysis, we find that most errors are caused by trigger extraction. The main reasons are as follows. First, not all of the event triggers are verbs, such as '婚姻 (marriage)' for the 'Life/Marry' event. This happens despite the fact that it is reasonable to assume that event triggers are verbs because on average more than 95% event triggers are verbs in the ACE 2005 corpus. Second, since only verbs with subject and object are extracted, non-predicate verbs and verbs without subject/object will not be extracted as candidate triggers. However, the coverage of possible triggers by our trigger extraction algorithm is reasonably good (more than 85%), because most of the trigger words appear repeatedly in the corpus, and their usages are varied. As long as one of their usages is fit for our extraction algorithm, they can be extracted as candidate triggers. Note that the goal of this study is to build an event type paradigm for new domains. We are concerned more with the coverage of the event type rather than event triggers. The event triggers extracted by us can cover all event types. We will explore a more effective trigger extraction algorithm in future work.

Trigger ambiguity also accounts for a large proportion of the errors. As discussed above, we cannot judge the event type only by the trigger itself, such as '撤 (withdraw/dismiss)' for both 'Personnel/End-position' event and 'Movement/Transport' event. This kind of error can be partially fixed by clustering

both triggers and participants. For example, we cluster '撤职务 (dismiss duties)' for a 'Personnel/Endposition' event and '撤军队 (withdraw troop)' for a 'Movement/Transport' event. These examples indicate that selectional preferences seem to be a reasonable feature even for highly ambiguous verbs like '撤 (withdraw/dismiss)', which encourages their use for the improvement of argument extraction.

There are still some errors caused by the trigger filter. This is mainly due to the fact that not all triggers are general or nominal verbs. More effective filter rules will be explored in the future.

Some other errors are caused by NLP tools, such as word segmentation, part-of-speech tagging, and dependency parsing. We believe that our algorithms can be improved with the improvement of these NLP tools. In addition, there are about 10% of good event triggers extracted but put into the wrong cluster by the trigger cluster.

4.2.5 Experiment with different values of the event seed clustering threshold

Different values of the threshold in Algorithm 2 can dramatically affect the performance of the event seed clustering. We experiment with different values of event seed clustering threshold to find the best value. Fig. 4 presents the effect on F-measure of varying thresholds. This figure shows that the best performance can be obtained by selecting the threshold 0.6 for the ACE corpus. Fig. 4 also suggests that the performance of seed clustering does not dramatically change with the volatility of the threshold from 0.5 to 0.8. Hence, we can first set the threshold at 0.6 for new domains.

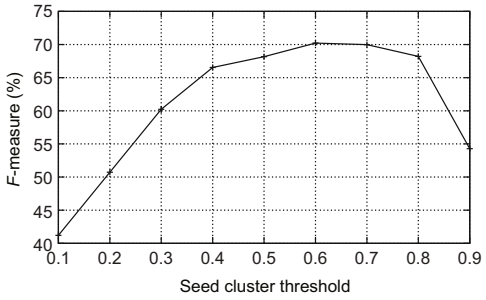


Fig. 4 Experimental results under different values of the event seed clustering threshold

4.3 Event type identification and argument recognition

Based on our event type paradigm, we proceed to implement traditional event extraction tasks, i.e., event type identification and event argument recognition. We adopt precision (P), recall (R), and F-measure (F) to evaluate the effectiveness of our approach, and compare it with a state-of-the-art event extraction system.

4.3.1 Baseline method

We use a state-of-the-art Chinese event extraction system, which was developed by Chen and Ji (2009), as our baseline. This system extracts events with an annotated corpus. Its training and testing procedures are as follows.

The system combines a word-based classifier with a character-based classifier. The event types are specified in advance. For every event mention in the ACE training corpus, features are extracted according to some language-specific issues. In addition, a set of maximum entropy based classifiers is trained:

1. Event type identification: to distinguish event mentions from non-event mentions, and to classify event mentions by type.

2. Event argument recognition: to distinguish event arguments from non-arguments.

In the testing procedure, each document is scanned for instances of triggers from the training corpus. If an instance is found by the trigger classifier, the system tries to assign some of the mentions in the sentence as arguments of a potential event mention. The argument classifier is applied to the remaining mentions in the sentence, for any argument passing that classifier; the role classifier will assign a role to it. Finally, the system will report the event with type and arguments.

4.3.2 Results and analysis

Table 6 shows the overall precision (P), recall (R), and F-measure (F) scores of the baseline system and our BUEES. The table also lists the performance of two human annotators from Chen and Ji (2009).

BUEES outperforms the baseline without annotated corpus. The performance of event type identification is 2.3% higher than that of the baseline system, and the performance of event argument recognition is 5.4% higher. Several conclusions can be drawn from Table 6:

Table 6 Overall experimental results

System/ Human	Type identification			Argument recognition		
	P	*R*	*F*	*P*	*R*	*F*
Baseline	65.7%	50.9%	57.4%	53.1%	36.2%	43.1%
BUEES	72.7%	50.7%	59.7%	51.3%	45.9%	48.5%
HA1	75.2%	74.6%	74.9%	58.6%	60.9%	59.7%
HA2	82.7%	80.3%	81.5%	66.8%	69.6%	68.2%

HA1: human annotator 1; HA2: human annotator 2

1. BUEES does not use any labeled corpus and achieves comparable performance with the baseline system.

2. Our approach on event type identification enhances the precision (by 7%) with little loss (0.2%) in recall compared to the baseline method. This recall loss is caused by the limited number of event seeds. The precision of event type identification is only 2.5% worse than that of one human annotator, which indicates that the precision of our approach is reasonably good. As our approach is pattern-based, its recall needs to be improved.

3. Our approach on event argument recognition enhances the F-measure (by 5.4%) compared to the baseline method. It indicates that our pattern learning algorithm is effective. We also find that the performance of human annotation on event argument recognition is not high enough (59.7% in F-measure), and hence it is a difficult problem requiring further study.

4. Table 6 also shows that the BUEES system improves the recall performance (by 9.7%) of event argument recognition over the baseline system. It shows that an unsupervised event extraction system can also achieve comparable or better performance compared with a supervised system. However, it is obvious that system recall is substantially lower than system precision. Although a bootstrapping-based approach can improve recall performance of the soft-pattern by increasing the number of iterations, when the number of iterations increases to a certain value, the recall will not be improved and the precision may decrease. It is mainly because the more the number of iterations, the more noise will appear in our system.

We should also note that human annotators use the perfect entity mentions, but our system extracts named entities automatically. So, the gap is also partially due to wrongly named entities.

5 Related work

Our system is designed to address two issues of event extraction: event type paradigm building and the traditional task of event extraction (i.e., event type identification and event argument recognition). The approach of event type paradigm building is related to some prior works on word cluster discovery (Barzilay and McKeown, 2001; Lin and Pantel, 2001; Ibrahim *et al.*, 2003; Hasegawa *et al.*, 2004; Miller *et al.*, 2004; Rosenfeld and Feldman, 2006). Most of these works are based on machine translation techniques to solve the paraphrase extraction problem. However, some recent research has stressed the benefits of using word clusters to improve the performance of information extraction tasks. For example, Miller *et al.* (2004) proved that word clusters could significantly improve English name tagging performance. In the same vein, some studies have addressed the problem of relation extraction (Poon and Domingos, 2008; 2009; Yates and Etzioni, 2009; Chambers and Jurafsky, 2009; 2011), where 'relation words' were extracted and clustered. Our work confirms that event seed clusters are also effective for event type paradigm building. The problem of event seed extraction and clustering is also a challenging problem.

The approach of event extraction is related to a weakly supervised pattern learning algorithm. Yangarber *et al.* (2000) used a bootstrapping based method learning simple surface patterns to extract information. Stevenson and Greenwood (2005) proposed similarity-centric bootstrapping which tries to find patterns with high lexical similarities. Liao and Grishman (2010) filtered ranking for bootstrapping in event extraction. They used two state-of-the-art bootstrapping-based event extraction systems and then ranked their candidate patterns and accepted the top-ranked patterns at each iteration. BUEES is similar to these systems in its general approach, but its surface patterns allow gaps that can be matched by any sequences of tokens, which makes the patterns much more general and allows the recognition of more instances compared with the simple surface patterns.

Some English event extraction systems based on pattern or machine learning have been reported (Yangarber *et al.*, 2000; Grishman, 2001; Patwardhan and Riloff, 2006; Ji and Grishman, 2008).

However, to our knowledge, non-English event extraction has rarely been reported. The baseline system is based on ACE Chinese events. Its contribution is to exploit language-specific features for Chinese event extraction. However, the reported precision of the results was lower than that of English event extraction. In contrast, the performance of BUEES, which is unsupervised, is better than that of the baseline system and as good as that of the state-of-the-art English system.

Web-scale IE has received considerable attention in the last few years. Pre-emptive IE and Open IE are the first paradigms that relax the restriction of a given vocabulary of relations and scale to all relation phrases (Shinyama and Sekine, 2006; Banko *et al.*, 2007; Banko and Etzioni, 2008; Wu and Weld, 2010; Etzioni *et al.*, 2011; Fader *et al.*, 2011). Pre-emptive IE relies on document and entity clustering, which is too costly for Web-scale IE. Open IE favors speed over deeper processing, which aids in scaling to Web-scale corpora. Compared with Pre-emptive IE and Open IE, the main differences of this study are: first, the previous work focuses mainly on relation extraction, but this study aims to extract events from a Web corpus; second, Open IE cannot give the event (or relation) type paradigm which is useful in application.

6 Conclusions and future work

We have presented the BUEES system which discovers interesting events, learns extraction patterns, and extracts the event instances from the Web. BUEES relies neither on manually produced extraction patterns nor on a manually annotated training corpus.

BUEES performs by clustering event seeds, generating seed instances, and bootstrapping its pattern. Based on a general sequential pattern, we propose a soft-pattern learning algorithm. Soft-pattern has a much greater generalizing ability and can reach a high performance for successful bootstrapping.

In the future we would like to explore better patterns. Patterns in this paper are very simple in that they just use several cost values and rules to generate a soft-pattern. We want to see if we can achieve higher performance with more complex and perfect patterns.

References

Ahn, D., 2006. The stages of event extraction. Proc. Workshop on Annotating and Reasoning about Time and Events, p.1-8.

Banko, M., Etzioni, O., 2008. The tradeoffs between open and traditional relation extraction. Proc. Annual Meeting on Association for Computational Linguistics, p.28-36.

Banko, M., Cafarella, M.J., Soderland, S., *et al.*, 2007. Open information extraction for the Web. Proc. 20th Int. Joint Conf. on Artificial Intelligence, p.2670-2676.

Barzilay, R., McKeown, K.R., 2001. Extracting paraphrases from a parallel corpus. Proc. 39th Annual Meeting on Association for Computational Linguistics, p.50-57. [doi:10.3115/1073012.1073020]

Chambers, N., Jurafsky, D., 2009. Unsupervised learning of narrative schemas and their participants. Proc. 47th Annual Meeting on Association for Computational Linguistics and 4th Int. Joint Conf. on Natural Language Processing, p.602-610.

Chambers, N., Jurafsky, D., 2011. Template-based information extraction without the templates. Proc. 49th Annual Meeting on Association for Computational Linguistics, p.976-986.

Che, W., Li, Z., Li, Y., *et al.*, 2009. Multilingual dependency-based syntactic and semantic parsing. Proc. 13th Conf. on Computational Natural Language Learning, p.49-54.

Chen, Z., Ji, H., 2009. Language specific issue and feature exploration in Chinese event extraction. Proc. Annual Conf. on Association for Computational Linguistics, p.209-212.

Chinchor, N., Lewis, D.D., Hirschman, L., 1993. Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3). *Comput. Ling.*, **19**(3):409-449.

Ding, X., Song, F., Qin, B., *et al.*, 2011. Research on typical event extraction method in the field of music. *J. Chin. Inform. Process.*, **25**(2):15-20 (in Chinese).

Ding, X., Qin, B., Liu, T., 2013. Building Chinese event type paradigm based on trigger clustering. Proc. Int. Joint Conf. on Natural Language Processing, p.311-319.

Dong, Z., Dong, Q., 2006. HowNet and the Computation of Meaning. World Scientific Publishing Company, USA.

Etzioni, O., Fader, A., Christensen, J., *et al.*, 2011. Open information extraction: the second generation. Proc. 22nd Int. Joint Conf. on Artificial Intelligence, p.3-10.

Fader, A., Soderland, S., Etzioni, O., 2011. Identifying relations for open information extraction. Proc. Conf. on Empirical Methods in Natural Language Processing, p.1535-1545.

Friedman, J.H., Bentley, J.L., Finkel, R.A., 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, **3**(3):209-226. [doi:10.1145/355744.355745]

Grishman, R., 1997. Information extraction: techniques and challenges. *In*: Pazienza, M.T. (Ed.), Information Extraction: a Multidisciplinary Approach to an Emerging Information Technology. Springer Berlin Heidelberg, New York, USA, p.10-27. [doi:10.1007/3-540-63438-X_2]

Grishman, R., 2001. Adaptive information extraction and sublanguage analysis. Int. Joint Conf. on Artificial Itelligence, Workshop on Adaptive Text Extraction and Mining.

Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2001. On clustering validation techniques. *J. Intell. Inform. Syst.*, **17**(2-3):107-145. [doi:10.1023/A:1012801612483]

Hasegawa, T., Sekine, S., Grishman, R., 2004. Discovering relations among named entities from large corpora. Proc. 42nd Annual Meeting on Association for Computational Linguistics, Article 415. [doi:10.3115/1218955.1219008]

Hirschberg, D.S., 1977. Algorithms for the longest common subsequence problem. *J. ACM*, **24**(4):664-675. [doi:10.1145/322033.322044]

Hong, Y., Zhang, J., Ma, B., *et al.*, 2011. Using cross-entity inference to improve event extraction. Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, p.1127-1136.

Ibrahim, A., Katz, B., Lin, J., 2003. Extracting structural paraphrases from aligned monolingual corpora. Proc. 2nd Int. Workshop on Paraphrasing, p.57-64. [doi:10.3115/1118984.1118992]

Ji, H., Grishman, R., 2008. Refining event extraction through cross-document inference. Proc. Association for Computational Linguistics, p.254-262.

Lee, C.S., Chen, Y.J., Jian, Z.W., 2003. Ontology-based fuzzy event extraction agent for Chinese e-news summarization. *Expert Syst. Appl.*, **25**(3):431-447. [doi:10.1016/S0957-4174(03)00062-9]

Liao, S., Grishman, R., 2010. Filtered ranking for bootstrapping in event extraction. Proc. 23rd Int. Conf. on Computational Linguistics, p.680-688.

Lin, D., Pantel, P., 2001. DIRT@SBT@discovery of inference rules from text. Proc. 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, p.323-328. [doi:10.1145/502512.502559]

Liu, T., Ma, J., Zhang, H., *et al.*, 2007. Subdividing verbs to improve syntactic parsing. *J. Electron. (China)*, **24**(3):347-352 (in Chinese). [doi:10.1007/s11767-005-0193-8]

Mei, J.J., Zhu, Y.M., Gao, Y.Q., *et al.*, 1983. Dictionary of Synonymous Words. Shanghai Dictionary Publishing Press, Shanghai, China (in Chinese).

Miller, S., Guinness, J., Zamanian, A., 2004. Name tagging with word clusters and discriminative training. Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, p.337-342.

Miwa, M., Sætre, R., Kim, J.D., *et al.*, 2010. Event extraction with complex event classification using rich features. *J. Bioinform. Comput. Biol.*, **8**(1):131-146. [doi:10.1142/S0219720010004586]

Pang, B., Knight, K., Marcu, D., 2003. Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, p.102-109. [doi:10.3115/1073445.1073469]

Patwardhan, S., Riloff, E., 2006. Learning domain-specific information extraction patterns from the Web. Proc. Workshop on Information Extraction Beyond the Document, p.66-73.

Pham, X., Le, M., Ho, B., 2013. A hybrid approach for biomedical event extraction. Proc. Association for Computational Linguistics, p.121-124.

Poon, H., Domingos, P., 2008. Joint unsupervised coreference resolution with Markov logic. Proc. Conf. on Empirical Methods in Natural Language Processing, p.650-659.

Poon, H., Domingos, P., 2009. Unsupervised semantic parsing. Proc. Conf. on Empirical Methods in Natural Language Processing, p.1-10.

Riloff, E., 1996. Automatically generating extraction patterns from untagged text. Proc. AAAI, p.1044-1049.

Ritter, A., Mausam, Etzioni, O., *et al.*, 2012. Open domain event extraction from Twitter. Proc. 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, p.1104-1112. [doi:10.1145/2339530.2339704]

Rosenfeld, B., Feldman, R., 2006. URES: an unsupervised web relation extraction system. Proc. COLING/ACL on Main Conference Poster Sessions, p.667-674.

Schilder, F., 2007. Event extraction and temporal reasoning in legal documents. *In*: Schilder, F., Katz, G., Pustejovsky, J. (Eds.), Annotating, Extracting and Reasoning about Time and Events, p.55-71. [doi:10.1007/978-3-540-75989-8_5]

Shinyama, Y., Sekine, S., 2006. Preemptive information extraction using unrestricted relation discovery. Proc. Conf. of the North American Chapter of the Association of Computational Linguistics on Human Language Technology, p.304-311. [doi:10.3115/1220835.1220874]

Soderland, S., 1999. Learning information extraction rules for semi-structured and free text. *Mach. Learn.*, **34**(1-3):233-272. [doi:10.1023/A:1007562322031]

Stevenson, M., Greenwood, M.A., 2005. A semantic approach to IE pattern induction. Proc. 43rd Annual Meeting on Association for Computational Linguistics, p.379-386. [doi:10.3115/1219840.1219887]

Sudo, K., Sekine, S., Grishman, R., 2003. An improved extraction pattern representation model for automatic IE pattern acquisition. Proc. 41st Annual Meeting on Association for Computational Linguistics, p.224-231. [doi:10.3115/1075096.1075125]

Wagner, W., Schmid, H., im Walde, S.S., 2009. Verb sense disambiguation using a predicate-argument-clustering model. Proc. CogSci Workshop on Distributional Semantics Beyond Concrete Concepts, p.23-28.

Wu, F., Weld, D.S., 2010. Open information extraction using Wikipedia. Proc. 48th Annual Meeting of the Association for Computational Linguistics, p.118-127.

Yangarber, R., Grishman, R., Tapanainen, P., *et al.*, 2000. Automatic acquisition of domain knowledge for information extraction. Proc. 18th Conf. on Computational Linguistics, p.940-946. [doi:10.3115/992730.992782]

Yates, A., Etzioni, O., 2009. Unsupervised methods for determining object and relation synonyms on the web. *J. Artif. Intell. Res.*, **34**(1):255-296.

Yeh, A., Hirschman, L., Morgan, A., 2002. Background and overview for KDD Cup 2002 task 1: information extraction from biomedical articles. *ACM SIGKDD Explor. Newslett.*, **4**(2):87-89. [doi:10.1145/772862.772873]