TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/17 pp251-258
Volume 15, Number 3, June 2010

Event Type Recognition Based on Trigger Expansion*

QIN Bing (秦 兵)**, ZHAO Yanyan (赵妍妍), DING Xiao (丁 效),
LIU Ting (刘 挺), ZHAI Guofu (翟国富)†

Research Center for Information Retrieval, School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China;
† School of Electrical and Information Engineering, Harbin Institute of Technology, Harbin 150001, China

Abstract: Event extraction is an important research point in information extraction, which includes two important sub-tasks of event type recognition and event argument recognition. This paper describes a method based on automatic expansion of the event triggers for event type recognition. The event triggers are first extended through a thesaurus to enable the extraction of the candidate events and their candidate types. Then, a binary classification method is used to recognize the candidate event types. This method effectively improves the unbalanced data problem in training models and the data sparseness problem with a small corpus. Evaluations on the ACE2005 dataset give a final *F*-score of 61.24%, which outperforms traditional methods based on pure machine learning.

Key words: event extraction; event type recognition; event trigger

Introduction

Event extraction is a challenging research point in information extraction. The goal of event extraction is to describe an event using natural language to give the time, place, and other participants and actions about an event. Event extraction can be used in many NLP application fields, such as automatic summarization[1-3], question and answering[4], and information retrieval[4].

There have been many event extraction studies[4-7]. The MUC (Message Understanding Conference) (1987-1998) played a significant role in the field of information extraction with event extraction always a key evaluation task. After the MUC, ACE (Automatic Content Extraction)[8] began to run event extraction tasks. The definition and corpus for event extraction used in this paper are from ACE2005[9].

An event is made up of an event trigger and event arguments, thus an event extraction task includes the following steps.

(1) Event trigger and event type recognition: An event trigger is a word starting an event, which is an important feature for recognizing the event type. ACE2005 defined 8 event types and 33 subtypes shown in Table 1. Event trigger and event type/subtypes ("event type" for short) recognition are the emphasis of this paper.

(2) Event argument recognition: Event arguments collectively refer to the event participants and event attributes. ACE set a template for every event type, each slot of which corresponds to the event argument role.

Figure 1 shows an example of an event. "出生" is the trigger word, its event type is "Life" and the subtype is "Be-Born". This event consists of three arguments, namely, "毛泽东", "1893 年", "湖南湘潭", which correspond to the three role labels in the Life/

Received: 2009-12-08; revised: 2010-01-21
* Supported by the National Natural Science Foundation of China (Nos. 60975055 and 60803093) and the National High-Tech Research and Development (863) Program of China (No. 2008AA01Z144)
** To whom correspondence should be addressed.
E-mail: bqin@ir.hit.edu.cn; Tel: 86-451-86413683

Be-Born event template of "Person", "Time-Within" and "Place", respectively.

Table 1 ACE event type and subtypes

Types	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

Event: 毛泽东 1893 年出生于湖南湘潭。
Event Type: Life
Event Subtype: Be-Born
Trigger: 出生
Arguments:
- ✧ Person: 毛泽东
- ✧ Time: 1893 年
- ✧ Place: 湖南湘潭

Fig. 1 An example of a "Born" event

There are two types of methods for event extraction. The pattern-based method generates an event pattern manually or automatically and uses various pattern matching algorithms to match the extracted events with the event patterns[7,10], e.g. the Yankova's soccer event extraction system[11] and the Lee's ontology-based meteorology event extraction system[12] and so on. The precision of the pattern-based method is very high. However, event patterns are difficult to identify for all cases due to the diversity of event expressions, which results in a low recall rate. In addition, the domain adaptiveness of the pattern-based method is very poor since event patterns are limited to a concrete domain. The machine learning-based method seeks to resolve the problem of domain independence. This method considers event extraction as a classification problem, which focuses on the classifier design and feature selection. Chieu and Ng[13] used a maximum entropy classifier for event argument recognition in event extraction in 2002. Ahn[4] combined MegaM with the Timbl machine learning method to recognize event types and event arguments in 2006. Event type recognition in Ahn's method used each word in a sentence as an example. Then, a binary classification method was used to determine whether the word was a trigger word, with a multi-classifier to recognize its event type. Ahn's method can automatically detect triggers and recognize the event types. However, it results in many negative examples since every word in the sentence is used as an example for training the machine learning model. Thus, it leads to a serious data unbalance between positive and negative examples. Moreover, the multi-classification suffers from data sparseness because of the small corpus.

The event type recognition in this paper is based on an automatic expansion of event triggers. The first step automatically extends the event triggers by a thesaurus to get candidate events (the sentence including the extended triggers) and their candidate types. The second step analyzes the trigger related features, such as the POS, context, and thesaurus information in a binary classification. The method uses only candidate events as the training examples, which effectively reduces the number of negative examples and resolves the data imbalance problem. The extended triggers are helpful to detect new events, while the binary classification avoids the low precision problem.

1 Automatic Expansion of Event Triggers and Candidate Event Extraction

An event trigger directly triggers an event, which is an important feature for determining the event type. The sentence containing the event trigger is called a candidate event. The paper then selects real events from these candidate events.

* 许洪敏患上了脑血栓，**瘫痪**在床。
(1) 工人工作期间不慎摔成**瘫痪**。
(2) 暴风雪突袭东北华北辽宁所有高速公路基本**瘫痪**。
(3) "府院之争"使政局陷入**瘫痪**。
(4) 黑客攻击导致网站**瘫痪**。

Fig. 2 Examples of candidate events

As shown in Fig. 2, sentence * is an event from the training corpus, its event trigger is "瘫痪", and event type is Life/Injure. All sentences containing "瘫痪" are candidate events and their candidate type is Life/Injure.

Real events can be identified using the candidate events and the candidate events' types can be directly determined from the corresponding event triggers. Hence, event trigger recognition is very important for event detection. However, the training corpus contains a limited number of events. Thus, the system may not discover similar but different wording events, and may lose some events. Consider the event "他偏瘫在床", where the event trigger "偏瘫" is not contained in the training corpus, so this event is not easily recognized. "偏瘫" and "瘫痪" have similar meanings, thus, a thesaurus can be used to automatically extend the event triggers to cover many more types of event triggers.

Candidate event extraction first automatically extends event triggers by a thesaurus; and then extracts candidate events based on the extended triggers.

1.1 Automatic expansion of event trigger based on a thesaurus

The algorithm first obtains seed triggers and their corresponding types, then uses a thesaurus to extend these triggers.

The event triggers labeled in the training corpus are called seed triggers. Seed triggers and their corresponding event types are then obtained from the training corpus, with each giving a binary pair (trigger, event type). Statistically, about 94% of the seed triggers can only trigger one event type, for example, (瘫痪, Life/Injure); while the other 6% of the seed triggers can trigger more than one event type, such as, "辞职", which belongs to both the Business/End-Org event type and the Personnel/End-Position event type. Since there is a low occurrence probability for a seed trigger with multiple types, the type with the greatest probability in the training data is used for the trigger. Then a "seed trigger-event type" binary pair table is built with Table 2 showing some examples.

Table 2 "Seed trigger-event type" binary pair table

Trigger	Event type
倒闭	Business/Declare-Bankruptcy
抗争	Conflict/Attack
逃离	Movement/Transport
示威	Conflict/Demonstrate
受伤	Life/Injure
访问	Movement/Transport
判刑	Justice/Sentence
丧生	Life/Die
…	…

Table 2 shows that each event trigger corresponds to only one event type with the event triggers in Table 2 covering all types of events.

To better deal with the new events, a thesaurus is used to extend the "seed trigger-event type" binary pair table. The thesaurus is the *TongYiCi CiLin* (expansion version) from the Harbin Institute of Technology, Center for Information Retrieval.

Ib16D01= **瘫痪** 风瘫 瘫 截瘫 偏瘫 半身不遂 脑瘫

Fig. 3 A fifth level example of *TongYiCi CiLin* (expansion version)

As shown in Fig. 3, the *TongYiCi CiLin* (expansion version) tags a group of words with the same sense, such as "瘫痪" in this group encoded as "Ib16D01=". Therefore, the seed trigger and the *TongYiCi CiLin* (expansion version) can be used to easily extend the "seed trigger-event type" binary pair table. The trigger expansion logic is not blind, but uses an appropriate threshold to limit the expansion as shown in Fig. 4.

Step 1:
For every seed trigger *t* in the "seed trigger-event type" table, find all its senses in *TongYiCi CiLin* (extension version)
Step 2:
If *n* or more than *n* words in a synset are contained in the "seed trigger-event type" table, and these words have the same event type, then all the words in the synset will be extended as triggers, and given the same event type of *t*. Here, *n* is called the expansion threshold.
Step 3:
Filter the extended triggers for multiple event types for the final "seed trigger-event type" binary pair table.

Fig. 4 Automatic trigger expansion algorithm

Take the seed trigger "关闭" as an example, which has two entries in *TongYiCi CiLin* (expansion version): "Fa31B01= 关 关闭 闭 合 掩 阖 密闭 虚掩 关掉 闭合 封关 闭锁" and "If22C02= 倒闭 关闭 停闭 停歇 关门 关张 关 关门大吉 闭馆". Assume an expansion threshold *n*=3 and analyze all the words in the synset "Fa31B01=". Since only "关闭" is in the "seed trigger-event type" table, the number is less than 3, thus this synset will not be extended. Similarly, synset "If22C02=" has three words in the "seed trigger-event type" binary pair table, namely, "关闭", "倒闭", and "关门大吉" , all of which correspond to the event type "Business/End-Org". Thus, other words such as "关张" in this synset are also used as triggers

with their event types as Business/End-Org.

Thus, this method effectively extends semantic words with the same event type by setting an expansion threshold, such as for synset "If22C02=" of Business/End-Org, and deletes noise words such as for synset "Fa31B01=".

1.2 Candidate event extraction based on the extended "seed trigger-event type" table

The extended "seed trigger-event type" binary pair table is then used for the extraction of candidate events and their candidate event types as shown in Fig. 5.

Step 1: Preprocess the article, including splitting the sentence and tokenization.
Step 2: For each sentence, find the words existing in the extended "seed trigger-event type" binary pair table.
Step 3: If a word w is found，the sentence with w is considered as a candidate event. If a sentence has many triggers, it contains many events.

Fig. 5 Candidate event extraction algorithm

This extraction process identifies many candidate events and annotates each candidate event with a possible type (the corresponding type of w in the extended "seed trigger-event type" binary pair table). This work lays a foundation for the binary classification of the candidate events.

2 Event Type Identification Based on Binary Classification

A large number of candidate events can be found using the automatic expansion of the event triggers. The extended "seed trigger-event type" table provides a possible candidate type for each candidate event. However, a larger number of the candidate events are not real events. As shown in Fig. 2, not all candidate events containing "瘫痪" are real Life/Injure events, only example 1 is a real event. Since examples 2-4 are not in line with the Life/Injure type, they are not real events and do not belong to any of the 33 ACE event types. Therefore, a classifier is needed to recognize real events.

This algorithm first finds possible candidate event types according to the extended "seed trigger-event type" table for each candidate event. Thus, the candidate event classification can be analyzed as a binary classification problem to determine whether a candidate event matches its candidate type.

2.1 Feature selection

Event type recognition has various basic linguistic features influencing the system performance. These features and their descriptions are as follows.

F_L Lexical features

- trigger: the event trigger for a candidate event;
- trigger POS: the lexical information for the event trigger.

An event trigger that triggers the event is the most direct reflection of the event type. This is a key part of the event type recognition. Clearly, the event trigger itself and its lexical information are important features.

F_C Context features

- p word POS tag on the left of the event trigger;
- p word POS tag on the right of the event trigger;
- q entity head word type on the left of the event trigger;
- q entity head word subtype on the left of the event trigger;
- q entity head word type on the right of the event trigger;
- q entity head word subtype on the right of the event trigger;

where p and q are integers, and $p,q \in [1,4]$.

The event type of a candidate event is determined by the semantics of the event trigger's context. Therefore, the context features are very important for event type recognition, including the context lexical features and the context entity head word type features from the ACE corpus. The entities included in the event reflect the arguments contained in the event. For example, a "Life/Be-Born" event usually contains "people", "time", "place", and other arguments, which are constituted by the entities. Therefore, the entity context information can be used to describe the meaning of the event trigger in the context and to draw distinctions with events that are not Life/Be-Born events.

F_T Thesaurus features

- This feature refers to the m layer label in *TongYiCi CiLin* (expansion version) for the event trigger.

The trigger label in *TongYiCi CiLin* (expansion version) reflects the meaning of the event trigger in the candidate event. Here, $m \in [1,5]$. If the event trigger is a polysemous word, all the semantic labels are listed.

Examples

This example describes in detail the various features for the event in Fig. 1, assuming p=4, q=1, and m=4.

$\boldsymbol{F}_{\mathrm{L}}$: The event trigger is "出生", and its corresponding POS is "v".

$\boldsymbol{F}_{\mathrm{C}}$: The four-word POS before trigger "出生" is (Null, Null, nh, nt) and after the trigger is (p, ns, ns, wp). The type for the entity "1893 年", which occurs before the trigger word "出生", is "Time" and the corresponding subtype is "Time". The type for the entity "湖南湘潭", which occurs after the trigger, is "LOC", and the corresponding subtype is "Region-General". If the POS and entities do not exist, then they are labeled as "Null".

$\boldsymbol{F}_{\mathrm{T}}$: The fourth layer label of the trigger "出生" is "Ib01B".

Therefore, the candidate event "毛泽东 1893 年出生于湖南湘潭。" has the feature set shown in Fig. 6.

1:出生 2:v 3:Null 4:Null 5:nh 6:ht 7:p 8:ns 9:ns 10:wp
11:Time 12:Time 13:LOC 14:Region-General 15: Ib01B

Fig. 6 Feature set for a candidate event

2.2 Maximum entropy-based binary classification

The maximum entropy model[14] is the theoretical basis for the Maximum Entropy classifier, which models all the known conditions and ignores unknown conditions. The classifier finds a probability distribution, which satisfies all the known facts and is not influenced by the unknown facts.

The key characteristic of the Maximum Entropy model is its independence hypothesis. Therefore, the useful features for the final classification can be freely increased without worrying about their negative influences. The classifier's training efficiency is also relatively high. These merits have enabled the Maximum Entropy model to be successfully applied in many natural language processing tasks, such as information extraction and syntactic analysis. Hence, the Maximum Entropy classifier is used here for the binary classification of the candidate events.

Many ingredients are considered when predicting whether a candidate event belongs to a candidate class. Assume that X is an event vector composed of these ingredients, and the variant y is true only if an event belongs to the candidate class. $p(y\,|\,X)$ denotes the probability of predicting a candidate event to be a candidate class, which will be estimated based on the maximum entropy. When $p(y\,|\,X)$ satisfies some certain constraint condition, the Maximum Entropy model maximizes the entropy defined as

$$H(p)=-\sum_{X,y}p(y\,|\,X)\log p(y\,|\,X) \tag{1}$$

Here, the constraint conditions denote all the known facts, which can be formalized as follows:

$$f_i(X,y)=\begin{cases}1, & \text{if } (X,y) \text{ satisfies the condition;}\\ 0, & \text{else.}\end{cases} \qquad i=1,2,\cdots,n$$

$f_i(X,y)$ are the features of the maximum entropy model, where n is the total number of features. The feature function $f_i(X,y)$ describes the relationships between the variants X and y. The final probability of the output is as

$$p^*(y\,|\,X)=\frac{1}{Z(X)}\exp\Big(\sum_i\lambda_i f_i(X,y)\Big) \tag{2}$$

where λ_i is the corresponding weight of the feature function $f_i(X,y)$, and

$$Z(X)=\sum_y\exp\Big(\sum_i\lambda_i f_i(X,y)\Big) \tag{3}$$

3 Experiments

3.1 Datasets and evaluation metrics

The ACE2005 corpus used for the experiments was from Broadcast news, newswire, and newspaper (633 documents in all). We randomly selected 473 documents as the training set, 80 documents as the development set and the remaining 80 documents as the test set. The ACE 2005 corpus is not only annotated with the entity and its head word attributes, but also with detailed information for each event, including the event trigger, event type, and event arguments.

The event type identification performance was evaluated with precision, recall, and F-score, respectively (represented as P, R, and F), which were computed in the standard manner.

$$F\text{-score}=\frac{2PR}{P+R} \tag{4}$$

$$P=\frac{\text{Number of events with correctly identified event type}}{\text{Total number of identified events}} \tag{5}$$

$$R=\frac{\text{Number of events with correctly identified event type}}{\text{Total number of standard events}} \tag{6}$$

3.2 Results and analysis

3.2.1 Experimental steps

For the trigger expansion-based event type identification method, four kinds of comparative strategies are designed.

- M_{seed}: All 621 seed triggers and their corresponding event types are acquired from the training set. Then tag all the events that contain the trigger words as real events with the trigger word types as the event candidate types. As in Fig. 2, "瘫痪" is a trigger, thus all the candidate events that contain "瘫痪" are treated as real events and their event type is "Life/Injure".
- M_{exSeed}: The thesaurus is used to extend the seed triggers. Then all events are tagged that contain the extended triggers as real events with the trigger word types as the event candidate types.
- $M_{Seed+ME}$: All events that contain the seed triggers are considered as the candidate events. Then, the Maximum Entropy binary classifier uses the lexical, context, and thesaurus features to classify the candidate events.
- $M_{exSeed+ME}$ (our method): All events that contain the extended triggers are considered as the candidate events. Then, the Maximum Entropy binary classifier uses the lexical, context, and thesaurus features to classify the candidate events.

In the experiments, the four parameters (p, q, m, n) estimated on the development data are found to be p=4, q=1, m=4, and n=4 for the optimal results. A maximum entropy (ME) modeling toolkit[15] is used for both $M_{Seed+ME}$ and $M_{exSeed+ME}$. The iteration parameter $i=35$ is tuned up on the development data.

The results in Table 3 show that:

- M_{exSeed} has significantly improved the recall of event type recognition than M_{seed}, increasing from 68.78% to 77.14%. That's because M_{exSeed} automatically extends the triggers, so more events can be recognized.
- Comparing the strategy $M_{seed+ME}$ with M_{seed}, and the strategy $M_{exSeed+ME}$ and M_{exSeed}, the precisions are increasing when adding the binary classification step. That's because the binary classification better models the candidate events by utilizing many useful features.
- The strategy $M_{exSeed+ME}$ achieves the best results, which illustrates the effectiveness of the trigger expansion.

Table 3 Performances of the various strategies for the development data set

Method	*R*/%	*P*/%	*F*/%
M_{seed}	68.78	38.60	49.45
M_{exSeed} (n=4)	77.14	32.36	45.60
$M_{seed+ME}$(p=4, q=1, m=4)	53.88	65.51	59.13
$M_{exSeed+ME}$ (p=4, q=1, m=4, n=4)	57.14	64.22	60.48

The experimental results on the test data using $M_{exSeed+ME}$ are shown in Table 4.

Table 4 Performance of $M_{exSeed+ME}$ on the test data

R/%	*P*/%	*F*/%
54.86	69.29	61.24

3.2.2 Choice of expansion threshold

Setting the expansion threshold is important for the final results. So this will be discussed in detail.

We select 621 seed event triggers from the training set and use *TongYiCi CiLin* (expansion version) to automatically extend the seed triggers. According to the different expansion thresholds, 7 groups of experiments are designed based on $M_{exSeed+ME,}$ and the results are shown in Table 5 and Fig. 7.

Table 5 Effect of different expansion threshold in $M_{exSeed+ME}$

Expansion threshold (n)	Number of triggers after expansion (num)	N_t	N_f	N_t:N_f
1	4115	2107	16 034	1:8
2	1392	2089	5080	2:5
3	923	2076	3856	5:9
4	804	2032	3635	4:7
5	717	1908	2798	2:3
6	649	1890	2476	5:7
0 (no expansion)	621	1892	2364	4:5

For each expansion threshold n, N_t is the number of candidate events annotated as "true" (positive examples), and N_f is the number of candidate events annotated as "false" (negative examples). $N_t : N_f$ denotes the ratio of the number of positive examples to the number of negative examples.

Table 5 shows how $N_t : N_f$ varies for the different expansion thresholds. When n=1, there is a wide gap between N_t and N_f. When $n \geqslant 6$, the number of extended triggers is too small, which makes the expansion useless. Therefore, we focus on the situation of n=2, 3, 4, 5. The performance curve is shown in Fig. 7.

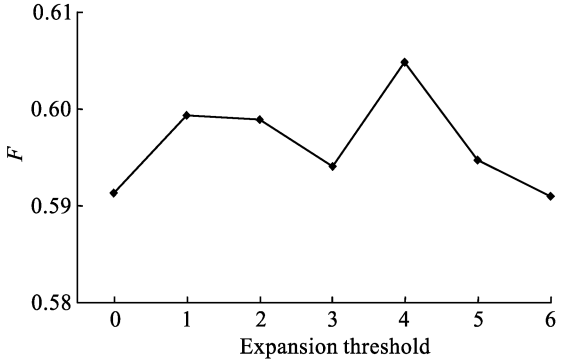

Fig. 7 Performance curve for different thresholds

Note that, n=0 refers to the $M_{seed+ME}$ method without the trigger expansion. The best performance is achieved for n=4 on the development set with F-score= 60.48%. It significantly performs better than $M_{seed+ME}$ with F=59.13%. Furthermore, n = 4 gives an appropriate number of extended triggers, and $N_t : N_f$ is balanced. Hence, n=4 is chosen in the experiments.

3.3.2 Contribution of the features on binary classification

The event trigger, which induces the start of an event, is known as a significant characteristic for determining the event class. The event trigger context and the thesaurus information are also used as classification features. The contributions of the three kinds of features are shown in Table 6.

Table 6 Contributions of three kinds of classification features

Method	R/%	P/%	F/%	Contribution value
$M_{exSeed+ME}(F_L+F_C+F_T)$	57.14	64.22	60.48	Standard value
No F_L	50.82	62.72	56.14	4.34%
No F_C	51.02	63.94	56.75	3.73%
No F_T	54.08	62.65	58.05	2.43%

In Table 6, No F_L, No F_C, and No F_T denote the experiment strategy without adopting F_L, F_C, and F_T, respectively. The difference between each result and the $M_{exSeed+ME}$ result is defined as the contribution value, which shows the contribution of the feature on $M_{exSeed+ME}$.

Obviously, all three kinds of features greatly influence the classification with lexical feature, F_L, having the greatest contribution. This result proves our assumption, event trigger information is very important for event identification. Consequently, the triggers are used to construct the candidate events.

3.3.3 Comparison with Ahn's method

In order to validate our method, we redo Ahn's idea[4] to be the baseline, which is a pure machine learning method with two key steps:

Step 1 Use every word in a sentence as a candidate trigger, execute binary classification, and then confirm whether the candidate trigger is a real event trigger.

Step 2 For all real event triggers, use a multi-classification algorithm to label the event class.

The feature selection is the same for $M_{exSeed+ME}$ (some features differ from Ahn's features, since there are always differences between Chinese and English corpora). In addition, the method uses the same evaluation method. The tuning of the maximum entropy parameter uses i=210 in the first step and i=30 in the second step to get the optimal result on the development set.

The test results are shown in Table 7.

Table 7 Comparison performances

	R/%	P/%	F/%
Ahn's method	38.91	52.36	44.64
Current method	54.86	69.29	61.24

We find in Table 7 that current method performs 16.6% better than Ahn's method. The concrete reasons are as follows.

(1) The recall of Ahn's method based on pure machine learning is very low. The main reason is that every word in a sentence is considered as a candidate trigger event when training the Maximum Entropy model. This leads to many negative examples and few positive examples. The experiments show that there are 132 793 training examples with N_t:N_f= 1:70. Thus, the data is not well balanced, which reduces the recall rate.

By comparison, our method uses the thesaurus to

automatically extend the triggers with only the sentences containing triggers as candidate events. In comparison with Ahn's method, the number of negative examples can be effectively reduced. The experiments show that there are 5667 training examples with $N_t : N_f$=4:7 (n=4). The data is well balanced. Furthermore, the extended triggers provide good coverage for all kinds of events. Thus, this method can better deal with new events, with an improve recall rate which is 14% higher than that of Ahn's method.

(2) The precision of Ahn's method is also not very high. The main reason is that there is the small corpus and the multiple classes, which make data very sparse.

By comparison, our method first judges whether an event belongs to a candidate class under the condition that the event is limited to the candidate class. This strategy can avoid the sparse data problem, which greatly improves the precision.

(3) Since the method based on pure machine learning has too many training and test examples, the efficiency is relatively slow. Our method only classifies the candidate event examples, so its efficiency is greatly improved.

4 Conclusions and Future Work

This paper presents an event type recognition method based on automatic trigger expansion. The main contribution lies in three aspects: (1) The *TongYiCi CiLin* (expansion version) is used to automatically extend event triggers, and further create candidate events. The extracted candidate events are used as training examples, which resolves the unbalanced data problem. In addition, this method can also identify new events, which improves the recall. (2) Lexical, context, and thesaurus information features are combined to descript a candidate event from various aspects, which avoid the low precision of multi-classification for small scale corpora. (3) The thesaurus information and machine learning algorithm are well integrated. Its final F-score achieves 61.24%, which is not only higher than that of a pure machine learning method, but also improves the efficiency.

Future work will identify more and better features to enhance the event type recognition. Event argument recognition is also an important part of event extraction and this will also be part of our research emphasis in future.

References

[1] Daniel N, Radev D, Allison T. Sub-event based multi-document summarization. In: Proceedings of the HLT-NAACL Workshop on Text Summarization. Edmonton, Canada, 2003: 9-16.

[2] Filatova E, Hatzivassiloglou V. Event-based extractive summarization. In: Proceedings of ACL Workshop on Summarization. Barcelona, Spain, 2004: 104-111.

[3] Li W J, Wu M L, Lu Q. Extractive summarization using inter- and intra- event relevance. In: Proceedings of the 44th Annual Meeting of the Association for Computational Liguistics. Sydney, Australia, 2006: 369-376.

[4] Ahn D. The stages of event extraction. In: Proceedings of the Workshop on Annotations and Reasoning about Time and Events. Sydney, Australia, 2006: 1-8.

[5] Zheng Chen, Heng Ji. Language specific issue and feature exploration in Chinese event extraction. In: Proceedings of NAACL HLT 2009. Boulder, Colorado, USA, 2009: 209-212.

[6] Heng Ji, Grishman R. Refining extraction through cross-document inference. In: Proceedings of ACL-08: HLT. Columbus, Ohio, USA, 2008: 254-262.

[7] Sasaki Y, Thompson P, Cotter P, et al. Event frame extraction based on a gene regulation corpus. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008). Manchester, Britain, 2008: 761-768.

[8] http://www.nist.gov/speech/tests/ace/. 2005.

[9] ACE (Automatic Content Extraction) Chinese Annotation Guidelines for Events. National Institute of Standards and Technology, 2005.

[10] Jiang J F. A research about the pattern acquisition for free text IE [Dissertation]. Beijing: The Institute of Computing Technology, the Chinese Academy of Sciences. 2004. (in Chinese)

[11] Yankova M. Focusing on scenario recognition in information extraction. In: Proc. EACL. Budapest, Hungary, 2003: 41-48.

[12] Lee C S, Chen Y J, Jian Z W. Ontology-based fuzzy event extraction agent for Chinese e-news summarization. *Expert Systems with Applications*, 2003, **25**(3): 431-447.

[13] Chieu H L, Ng H T. A maximum entropy approach to information extraction from semi-structured and free text. In: Proceedings of the 18th National Conference on Artificial Intelligence. Edmonton, Canada, 2002: 786-791.

[14] Berger A L, Della Pietra S A, Della Pietra V J. A maximum entropy approach to natural language processing. *Computational Linguistics*, 1996, **22**(1): 39-71.

[15] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html. 2009.