Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080341

Event Detection in Multiple Webpages based on Comprehensive Dimension Matching and Co-occurrence Constraint

*Yuanzi Xu, Qingzhong Li**, *Zhongmin Yan and Wei Wang*

School of Computer Science and Technology, Shandong University, Jinan, 250101, China

Received: 12 Jun. 2013, Revised: 17 Oct. 2013, Accepted: 18 Oct. 2013
Published online: 1 May. 2014

Abstract: Detecting various sentence-level events from multiple webpages can be important in finding knowledge. We propose an event detection method based on comprehensive dimension matching and co-occurrence constraint. First, we detect events from a single webpage by clustering co-reference sentence-level event mentions. These events are considered as co-occurrence events in every single webpage. Second, similar events from multiple webpages are clustered. The dimension matching method is used to aggregate event mentions. Different matchers measure different dimensions, and an extended evidence theory is proposed to allocate dynamic weight and combine dimension measurement results. We propose an event co-occurrence constraint to reduce match times and quantity of candidate matches events in the multiple webpages event-detection process to improve event cluster efficiency. The experiment results demonstrate that this method can detect various events and noticeably reduce the quantity of co-reference events.

Keywords: Event detection, comprehensive dimension matching, co-occurrence constraint, extended evidence theory

1 Introduction

Event detection is the task of identifying events in news. Most of events which are reported in webpages are unstructured data. Information management not only integrates structured data such as entity attributes and relationships between entities in the Web but also integrates events. Existed event detection approaches are inapplicable for discovering events that are participated by specific entity and the type of them cannot be predefined. Examples of such events include enterprise activities in the news. Integrating multiple enterprise webpages and detecting valuable entity events may help enterprise policymakers understand themselves and the development of novel trends in other enterprises, as well as improve market intelligence. An event is an activity that occurs at a special time and involves participants. An event mention is the sentence in which an event is reported. The descriptions of the same event from different webpages are inconsistent. These event mentions that report the same event are co-reference event mentions. If each event mention is considered an event, many duplicate events will occur. Thus, clustering co-reference event mentions as an event can reduce the number of duplicate events. Our goal is to detect events in multiple webpages by clustering co-reference event mentions. This approach is important to provide valuable market information to enterprises.

Detecting such events in multiple webpages poses several interesting technical challenges. First, event mentions are unstructured data, and the direct use of attribute similarity to judge entity attributes is difficult. For example, Fig. 1 depicts that an event has four co-reference event mentions. However, the texts of these co-reference event mentions have obvious differences. We use several dimensions to express event mentions and use comprehensive matchers to measure the similarities of these dimensions. Second, we find some events in single webpage by clustering co-reference event mentions. These events have obvious differences and can be considered as co-occurrence events. Co-occurrence events do not need to calculate the similarity between them and they can be used to reduce event match times in following event detection. Third, clustering event mentions in multiple webpages require considerable running time. We

* Corresponding author e-mail: lqz@sdu.edu.cn

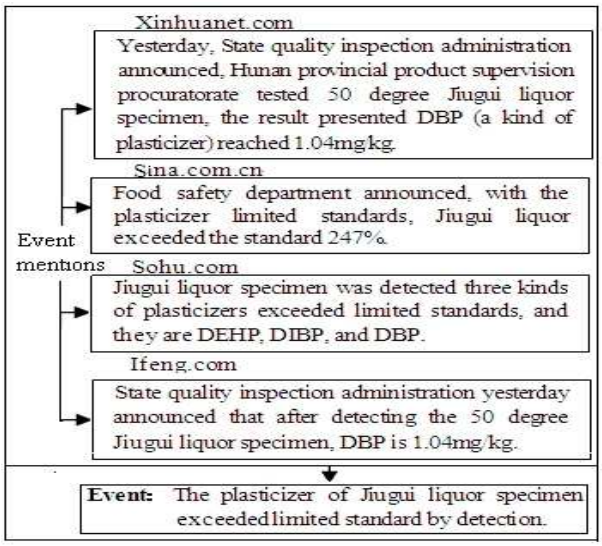

Fig. 1: Four co-reference event mentions for same event

propose the two-stage event detection method and the use of a constraint to reduce the running time.

Fig. 1 illustrates an event that has four co-reference event mentions. The event mention in Xinhuanet.com reports the plasticizer name. The event mention in Sina.com.cn reports the percentage of plasticizer that exceeds the limit standard. The event mention in Sohu.com introduces the three plasticizers. These co-reference event mentions are extracted from different webpages in the same day. However, these co-reference event mentions only use text similarities cannot find references to the same event. To improve event cluster accuracy, we adopt frame semantics [1,2] and use dimensions such as time, subject, object, and activity to represent event mention. First of all, we measure the similarity of each dimension. Thereafter, we use extended evidence theory to combine the similarities of key dimensions as the similarity of the final event mention. Considering that composite dimension matching can maximize the use of syntactic structure information and semantic information, this method can increase the recall and precision ratio of event detection.

We present a two-stage sentence-level event detection method, i.e., EDCoAGENES, to cluster co-reference event mentions in multiple webpages. Given a time period and a set of entities, we label event mentions that contain target entities in every webpage. In a single webpage, we cluster co-reference event mentions and find co-occurrence events. In multiple webpages, we use co-occurrence constraint to reduce match times and quantity of candidate matches events and we cluster similar events to detect many different events. By using the EDCoAGENES method, we not only detect various events but also can understand the event burst degree during a particular time period according to the amount of co-reference event mentions.

In summary, we introduce a two-stage sentence level event detection method to discover multiple webpage events. In this method, the second stage can use the co-occurrence events which are found in the first stage to cluster events. We use several dimensions to present event mentions and use comprehensive matchers in measuring event mentions to improve cluster accuracy. In this paper, we propose the use of event co-occurrence constraint to restrict the event detection of multiple webpages and reduce the running time during the cluster process.

This paper is organized as follows. We briefly review some related research efforts in Section 2, and describe the problem in Section 3.The proposed approach is introduced in Section 4 and the experimental evaluations are reported in Section 5. In the last section, we draw conclusions.

2 Research Background

In recent years, event detection research has increasingly focused on large-scale webpages and has become the most popular research in information management [3], data integration, information retrieval, and artificial intelligence. Information retrieval identifies similar changes in the relationship [4] between certain entities, such as simultaneously rise and decline, to detect events. Social tagging [5,6] discovers burst tags in the same period to detect a few special events. Manoj [7] found real-time events according to high frequency words in micro-blogs reported by the same author. Detecting many retrospective events in multiple webpages is important for finding knowledge. Event detection research involves single webpage detection and multiple webpages detection.

In single webpage event detection, Yan [8] found paragraph boundary events and considered them interrelated and independent. We propose the event co-occurrence constraint which is based on this idea of the independent relationship between events. In multiple webpage event detection, some studies have adopted predefined-type event detection. Shan [9] used event keywords to detect burst events. Zhao [10] appended time characteristics to event keywords, such as “2008 US presidential election”, which detects time-characteristic burst events. Given that detailed event information is needed to mine sentences in a webpage, current research has now focused on sentence-level event detection. Martina [11] proposed sentence-level event detection in news webpages and found predefined type events. Jiang [12] used open information extraction and proposed event ontology to find various events. David [13] thought important named entities, such as time, place, people, and organization, and the co-occurrence relationships between them were also used to find events. Cluster method is important for event detection. Compared with another event cluster method [14,15], the agglomerative

hierarchical cluster method [16] possesses certain advantages, such as accurate differentiation of objects, automatic determination of the number of clusters, and discovery of arbitrary clustering shapes. Thus, the agglomerative hierarchical cluster method is suitable for event detection.

Considering that many co-reference event mentions have missing and inconsistent values, the clustering of sentence-level event mentions according only to text similarity judgment exhibits low accuracy. Topic detection and tracking research (TDT) [17] uses the TFIDF method to calculate event keywords and uses keyword similarity as event similarity. Given that the same keywords in different syntactic structures, such as subject or object, can express different events, the use of syntactic structure can improve cluster accuracy. Li [18] demonstrated that comprehensive similarity can improve clustering precision for complex data structures. Evidence theory [19] combines several dimension similarities with static weight to provide a uniform result.

Our research used eight dimensions to present event mentions and combined syntactic structures and keyword semantic meanings to measure event mention similarity. We extended the evidence theory to allocate dynamic weights to different dimensions according to the ability of such dimensions to provide similarity.

3 Event Mention and Event

To present and facilitate the following discussions clearly, we explain some concepts used in this paper in this section.

Event mention. An event mention is a sentence in which an event is reported. An event can have many mentions that refer to it. In this paper we use eight dimensions to represent event mentions according to the definition of an event. The eight dimensions are denoted as { *agent*, *activity*, { *object* }, *time*, { *location* }, { *cause* }, { *purpose* }, { *manner* }}. The event mention set is represented as $EM = \{em_{11}, em_{12}, \cdots, em_{i1}, \cdots, em_{ij}, \cdots, em_{nk}\}$, and $em_{ij}(1 \leq i \leq n, 1 \leq j \leq k)$ represents an event mention in event e_i.

Event. An event is an entity activity that occurs at a specific time and place. An event is constructed by certain elements, such as time, agent, activity, object, location, cause, purpose, and manner. An event set is indicated as $E = \{e_1, e_2, \cdots, e_n\}$, and an event is expressed as $e_i(1 \leq i \leq n)$.

A data source provides event mentions, such as web sites, databases, etc. A set of data sources can be represented as $S = \{s_1, s_2, \cdots, s_n\}$.Some event mentions can point to the same event or to different events. The relationship between source, document, event mention, and event is shown in Fig 2.

Event Detection. Event detection is the automatic identification and classification of co-reference event mentions to find various events in mutiple webpages. Event detection is divided into single webpage event detection and mutiple webpages event detecion. Detected events from a single webpage construct co-occurrence events and they can simplify event detection in mutiple webpages. Event e_i can be represented as $e_i = \{em_{i1}, em_{i2}, \cdots, em_{in}\}$, and event set can be represented as $E = \{e_1, e_2, \cdots, e_i\} = \{\{em_{11}, em_{12}, \cdots, em_{1k}\}, \cdots, \{em_{i1}, em_{i2}, \cdots, em_{in}\}\}$. k, p and n are the number of co-reference event mentions.

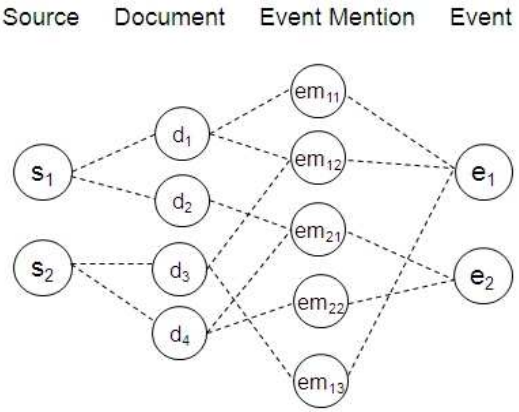

Fig. 2: Source, document, event mention, event, and their relationships

Co-occurrence Event. Events clustered from a single webpage are independent events which have obvious differences. Independent events from a single webpage are co-occurrence events. Co-occurrence events need not to calculate similarity in mutiple webpages event detection becuse they are obviously different. For example, if $\{e_1, e_2, \cdots, e_n\} \subseteq d_i$then $\{e_1, e_2, \cdots, e_n\}$ constitute co-occurrence event.

Event co-occurrence constraint. The event co-occurrence constraint restricts co-occurrence events to eliminate the need to calculate the similarity between co-occurrence events. Events that are detected in the same webpage constitute co-occurrence events. Co-occurrence events are defined as E_d. and d is the document in which we find some events. Two events from different webpages constitute an event pair. If the similarity of an event pair reaches the threshold, the event similarity in the matched event pair do not need to be computed with the co-occurrence event similarity.

4 Event Detection

4.1 Event detection process

In this paper, we present an event detection method named EDCoAGENES based on comprehensive dimension matching and co-occurrence constraints. This method can detect many events and reduce duplicates in event quantity. We use a slide window mechanism

according to the time period to obtain webpages. The whole event detection process is shown in Fig.3. The three steps involved in the event detection process are introduced as follows.

Step 1. An event mention is labeled in the webpages. The event mention that contains the target entities is then labeled by using related webpages. We use the Chinese word segmentation tool ICTCLAS 3.1 to handle event mention and use eight dimensions to represent the event mention. We filter the content of the webpage, label the event mention, and use dimensions to represent the event mention.

Step 2. An event is detected in a single webpage. In this step, we identify co-reference event mentions and cluster them in a single webpage. These events which are clustered from a single webpage are co-occurrence events. For example, the event mentions em_1 and em_2 in D_1 both refer to event e_1 (Fig. 3). These event mentions are co-reference event mentions that need to be classified in a cluster. Four event mentions in D_1 are classified into two clusters, which are referred to as event e_1 and e_2. Event e_1 and e_2 are co-occurrence events because they are detected in the same webpage. To compute for event mention similarity, we use the comprehensive dimension matching method and extended evidence theory model to allocate dynamic dimension weights. We classify the single webpage co-reference event mentions in a cluster as an event and do not merge them.

Step 3. An event is detected in multiple webpages. We pair any two webpages to cluster similar events and place the resulting cluster in a document. After that, we pair the document with another webpage to cluster similar events. For example, we pair D_1 and D_2 to cluster events, place the results in a document, and pair this document with D_3. In this process we find that e_1 and e_3 are similar events, and that e_4 and e_5 are similar events. We propose an event co-occurrence constraint to restrict clusters in a webpage pair. Event e_1 and e_2 are co-occurrence events in D_1, and events e_3 and e_4 are co-occurrence events in D_2. In the matching process we find that the similarity of e_1 and e_3 reached the threshold. We then aggregate e_1 and e_3 and do not compute the similarity of e_1, e_4, and e_2, e_3 because they are subjected to the event co-occurrence constraint.

Fig. 3 shows the three steps of event detection. We use the event detection algorithm to cluster co-reference event mentions in multiple webpages. The core algorithm is presented in Algorithm 1.

Algorithm 1 first initializes event set E and every document d_i from set D in Line 1. Lines 2 to 5 show the process of single webpage event detection. Line 3 labels the event mention EM_d from each document d_i, and Line 4 uses a function to aggregate event mentions if their similarity is greater than T. Lines 6 to 8 show event detection in multiple webpages. We pair two event sets to compare event similarity until all event sets are compared. AggregateEvent is a function that clusters similar events from two event sets and will be presented later.

Algorithm 1 Event Detection

Input: webpage set D. Similarity threshold T
Output: Event set E,every event is a cluster of co-reference event mentions

1. $E = \emptyset, d_i \in D$
2. for each document d_i from D
3. EM_d = LabelOneDocumentMention(d_i)
4. E_i = AggregateEventMention(EM_d)
/* aggregate event mentions if their similarity is greater than T */
5. end for
6. $E = E_1$
7. for $i = 2$ to $|D|$
8. $E = E \bigcup$ AggregateEvent(E, E_i)
/* aggregate two events if their similarity is greater than T */
9. end for
10. return E

4.2 Event detection in a single webpage

Event detection in a single webpage uses the hierarchical clustering method to cluster co-reference event mentions that refer to the same event because co-reference event mentions differ from literal descriptions. In clustering process we use dimension matching method. We use eight dimensions to represent event mentions to combine syntactic structures and text similarities for event mention comparisons. We find that the time, subject, object, and activity dimensions are usually not blank and other dimensions are usually blank or do not conform to other corresponding dimensions. According to event definition, time, activity and participants are key elements in an event. In this paper we divided participants into subject and object by different roles. So time, activity, subject and object are key dimensions which can basically represent an event. In this stage, we choose the time, subject, object, and activity dimensions as key dimensions in comparing event mention similarity.

We select event mentions in different webpages to explain dimension matching in Fig. 4. The time dimension value is the time the event occurred according to when the webpage reported the event and the specific time it appeared in the news. We also use a conjecture event occurrence time by using an offset value from the news, such as yesterday and two days ago. The activity dimension value is an activity verb that is extracted from a phrase by using the shallow semantic parsing method. We calculate the similarity of each dimension, and use extended evidence theory to compute for comprehensive dimension similarity. After clustering co-reference event mentions in a single webpage, we consider events which are detected in the same webpage as co-occurrence events.

In Fig.4, the first three event mentions refer to the same event even though these event mentions have noticeable differences in their literal descriptions. The key dimension

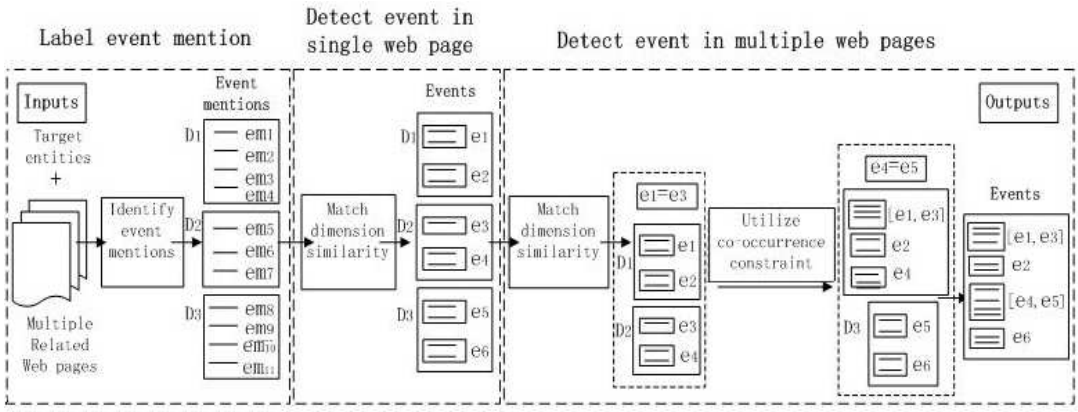

Fig. 3: Event detection process in multiple webpages

Event mention	Key dimension value
Food safety department announced, with the plasticizer limited standards, Jiugui liquor exceeded the standard 247%.	2012.11.19 | Jiugui liquor | plasticizers | exceeded
Jiugui liquor specimen was detected three kinds of plasticizers exceeded limited standards, and they are DEHP, DIBP, and DBP.	2012.11.19 | Jiugui liquor | plasticizers | detected
State quality inspection administration yesterday announced that after detecting the 50 degree Jiugui liquor specimen, DBP is 1.04mg/kg.	2012.11.19 | 50 degree Jiugui liquor | DBP | detecting
After Jiugui liquor company published the announcement of plasticizers, stock (code 000799) temporary cease trading in November 19, 2012.	2012.11.19 | company | stock | cease trading

Fig. 4: Key dimensions of event mentions

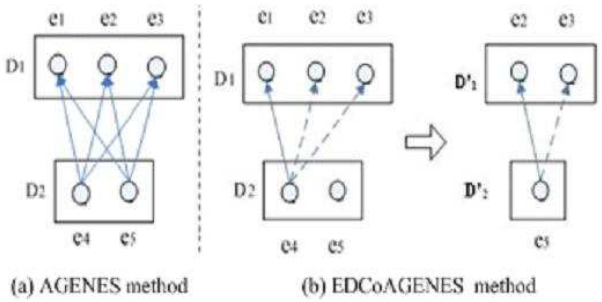

Fig. 5: Comparisons of co-occurrence constraint

values of the event mentions show slight differences. The last event mention refers another event, and we can clearly distinguish such an event according to the key dimension. In this paper, we detect events in a short time period (i.e., one week).

4.3 Event detection in multiple webpages

Event detection in multiple webpages pairs two webpages and uses the hierarchical clustering method to aggregate similar events. In this paper, we use event co-occurrence constraint to reduce event match times and quantity of candidate matches events in multiple webpages cluster process.

We note that co-occurrence events do not need to calculate similarity according to the constraint. Co-occurrence events can quickly cluster similar events in webpage pairs. We propose two rules to reduce match times and quantity of candidate matches events by using the event co-occurrence constraint.

Rule 1. If the similarity of an event pair reaches the threshold, the comparison process can be terminated. For example, if $e_i \in E_1$ and $e_j \in E_2$ construct a matched event pair $< e_i, e_j >$, the comparison process, which searches for an event similar to e_i, is terminated. Because the candidate matches events in E_2 are co-occurrence events of e_j. This rule can be used to reduce match times in event comparison process.

Rule 2. If a matched event pair in two webpages is found, the two events can be deleted from the candidate match event set. For example, for a matched event pair $< e_i, e_j >$, $e_i \in E_1$, and $e_j \in E_2$, the remainder events in E_1 are the co-occurrence events with e_i , and a matched event pair cannot be constructed with e_j. Thus, the candidate matches event sets are $\{E_2 - e_j\}$ and $\{E_1 - e_i\}$. This rule can be used to reduce the number of candidate matches events in event comparison process.

Fig. 5 show the difference in the comparison process of the agglomerative hierarchical clustering method (AGENES) and EDCoAGENES.

Fig. 5(a) shows the process used in the AGENES method if $< e_1, e_4 >$ is the matched event pair that compared e_4 with the remaining events in D_1. Fig. 5(b) presents how the EDCoAGENES method is used if the

Algorithm 2 Aggregate Event

Input: Event set E_k, E_p from document d_k and d_p. Similarity threshold T

Output: Event set E_{new}, every event contains co-reference event mentions from two webpages

```
1.  for each event e_ki from E_k
2.    for each event e_pj from E_p
3.      Sim(e_ki, e_pj);
      /* computer similarity of event e_ki and e_pj */
4.      if Sim(e_ki, e_pj) < T
5.        j++;
6.      else
7.        e_kipj = e_ki ∪ e_pj
        /* use co-occurrence constraint to stop match
        and aggregate e_ki, e_pj as e_kipj */
8.        E_new = E_new ∪ {e_kipj};
9.        E_p = E_p − {e_pj};
10.       E_k = E_k − {e_ki};
11.     end if
12.   end for
13.   i++;
14. end for
15. return E_new
```

matched event pair is $< e_1, e_4 >$. Comparisons for finding similar events at e_4 can be stopped. Thereafter, e_1 from D_1 and e_4 from D_2 are deleted. Dashed arrows point to the unnecessary comparisons in Fig.5(b). Given that e_1, e_2, and e_3 are co-occurrence events in D_1, and e_4, e_5 are co-occurrence events in D_2, we use the constraint only when we need to compare the similarity of e_5 and e_2. The left part of Fig.5(b) shows that we can reduce match times of e_4 by rule1. According to rule2, only e_2 and e_3 are candidate matches events of e_5. Reducing the number of candidate matches events is shown in the right part of Fig.5(b).

Algorithm 2 shows how constraint restricts clustering to aggregate similar events in two webpages.

Lines 1 and 2 circularly compare the events between E_k and E_p. Line 3 computes for the similarity between e_{ki} and e_{pj}. If the similarity is below the threshold T, the similarity of the next event in E_p with e_{ki} is computed. Lines 7 and 8 use an event co-occurrence constraint to aggregate e_{ki} and e_{pj} if their similarity reaches T. Lines 9 and 10 delete e_{pj} from E_p and e_{ki} from E_k by co-occurrence constraint.

We propose that the use of event co-occurrence constraint reduce running time. We assume that event set E_1 has m events and event set E_2 has n events. AGENES compares each event to all events in another set (Fig.5) and the time complexity is $O(m+n)^2$. EDCoAGENES finds two events that have reached the threshold and then stops the comparison process to delete the two events from their respective sets. For example, event e_i from E_1 is matched with event e_i from E_2. The match is stopped and e_i from E_1 and e_i from E_2 are deleted. The candidate match event quantity is $(m+n-2)$. If each iteration can find a matched event pair, the quantity of matching events decreases regularly as an arithmetic progression. Compared with AGENES, EDCoAGENES divides cluster events into two stages, within the document (single webpage) and across documents (multiple webpages). The comparison process in the first stage is similar to AGENES, and the time complexity is $O(m)^2$. Given that the event quantity in one document is small, the running time is short. The second stage uses a constraint to reduce match times and quantity of candidate match events to reduce running time, and the time complexity is $O(m \cdot log_2 m)$. In event detection in multiple webpages, EDCoAGENES is much faster than AGENES.

4.4 Similarity measure method

In this paper, we propose clustering co-reference event mentions and similar events by comprehensive dimension matching. This method combines syntactic structure information and literal similarity to improve clustering accuracy. We use eight dimensions to represent an event mention and use three matchers to measure the similarity of the key dimensions. In this section, we introduce the three matchers and use extended evidence theory to combine the dimension measurement results as a similarity of an event mention.

4.4.1 Similarity measure matcher

We introduce three matchers to compute the similarity among the time, subject, object, and activity dimensions.

1. Time value matcher

The Time value matcher is used to measure the time dimension. This matcher is adapted for identical or compatible data types. The default time value is the "date" type. If the time in which the event occurred is precise, the time can be extended to the "date-time" type. The numerical difference between two time data is the time distance, which is multiplied by a normalized factor that can obtain time dimension similarity.

$$Sim_{Num}(T_1, T_2) = \frac{logD}{|T_1 - T_2|} \quad (1)$$

$|T_1 - T_2|$is the absolute value of the time numerical difference, and $logD$ is the normalized factor . We set $D = 20$ by experiment.

2. Morpheme matcher

Morpheme refers to the independent and basic concept, such as indivisible word. This matcher is literally indivisible. The morpheme matcher combines text similarity and word order similarity for applications to Chinese phrase characters. We add larger weights to words that appear later in the word order. For example, the subject dimension value is "50 degree Jiugui liquor". The weight of "Jiugui liquor" is larger than the weight of "50 degree". In an event mention pair, N_a and N_b are two

subjects that need to be compared. The word orders are $N_a = \{a_i | i = 1,2,\cdots,m\}$ and $N_b = \{b_j | j = 1,2,\cdots,n\}$. Thus, the similarity of N_a and N_b is expressed as follows:

$$Sim_{Mp}(N_a, N_b) = \frac{2}{1/\sum \frac{a_i}{\sum N_a(i)} + 1/\sum \frac{b_j}{\sum N_b(j)}} \quad (2)$$

The formula $\sum \frac{a_i}{\sum N_a(i)}$ shows the comprehensive weight of the same morpheme N_a and N_b contained according to the location in N_a. $\sum \frac{b_j}{\sum N_b(j)}$ shows the comprehensive weight of the same morpheme N_a and N_b contained according to the location in N_b. The morpheme matcher can compute the similarity of the subject and object dimensions.

3. Semantic matcher

Considering that accurate reasons can be obtained by comparing verbs and verb phrases by semantic similarity, we use the semantic matcher method and the Hownet semantic network architecture [20] to measure verb similarity. In the Hownet semantic network architecture, a word is composed of primitives. The activity verb itself may be the primitive or can be deconstructed into primitives. We use primitive $S_a = \{a_i | i = 1,2,\cdots,m\}$ to denote verb a, $S_b = \{b_j | j = 1,2,\cdots,n\}$ to denote verb b, and adopted two semantic matchers to compute verb similarity.

(1) Path-based semantic matcher

The path-based semantic matcher calculates the similarity between two primitives by the path length of the two primitives.

$$Sim_{PSe}(S_a, S_b) = \frac{\alpha}{\alpha + dist(a_i, b_j)} \quad (3)$$

In this formula $dist(a_i, b_j)$ is the shortest path in Hownet, and α is an adjustment parameter that is the distance value when word similarity is 0.5. The path length is inversely proportional to similarity.

(2) Depth-based semantic matcher

The depth-based semantic matcher calculates the similarity between two primitives by the depth of their common ancestor.

$$Sim_{DSe}(S_a, S_b) = \frac{2 \times depth(a_i, b_j)}{depth(a_i) + depth(b_j)} \quad (4)$$

In this formula $depth(a_i, b_j)$ is the whole depth of the common ancestor of primitives in Hownet, $depth(a_i)$ and $depth(b_j)$ denote the depth of each primitive with a common ancestor.

The following is the composite semantic similarity formula:

$$Sim_{Sem} = \beta_1 Sim_{Pse}(S_a, S_b) + \beta_2 Sim_{Dse}(S_a, S_b) \quad (5)$$

In this formula $\beta_1 + \beta_2 = 1$ and we set $\beta_1 = 0.55$, $\beta_2 = 0.45$ by experiment.

We use the maximum primitive similarity formula to calculate the similarity of two verbs:

$$Sim_{Sem}(a,b) = max_{i=1,2,\cdots,m, j=1,2,\cdots,n} |Sim_{Sem}(S_a, S_b)| \quad (6)$$

4.4.2 Use of extended evidence theory to combine dimension matching

After obtaining the similarity of dimension value, we need to find a model to combine the dimension results as the total similarity of an event mention. Evidence theory [21] combines some evidence to obtain an objective and comprehensive result such that the model can be used to calculate the fusion result of multiple objects. In the evidence theory model, the recognition framework Θ contains all subsets, and the set of all subsets is presented as 2^Θ. The evidence theory model defines the probability distribution function as the map $2^\Theta \to [0,1]$, which is based on 2^Θ. The probability distribution function satisfies the conditions of $m(\phi) = 0$ and $\sum_{A \in 2^\Theta} m(A) = 1$. In the conditions, $\emptyset$ is null and A is any subset. The fusion formula of the evidence theory model is as follows:

$$m(A) = \frac{\sum_{\cap A_i = A} \Pi_{1 \le i \le n} m_i(A_i)}{1-K}, K = \sum_{\cap A_i = \emptyset} \Pi_{1 \le i \le n} m_i(A_i) \quad (7)$$

In this formula, we observe that each dimension provides the same contribution degree to the final result. However, the evidence characteristics of different dimensions are not the same. For example, we extract event mentions in webpages that are reported in the same day. In this situation, the similarity of the subject dimension is more important than the time dimension. To solve this problem and obtain a more accurate value for the similarity of the event mention, we extend the evidence theory model by allocating dynamic weight factor W_i for each dimension. We add a dynamic weight factor in each probability distribution function, and the fusion formula of extended evidence theory model is as follows:

$$m(A) = \frac{\sum_{\cap A_i = A} \Pi_{1 \le i \le n} W_i[m_i(A_i)]}{\sum_{\cap A_i \ne \emptyset} \Pi_{1 \le i \le n} W_i[m_i(A_i)]} \quad (8)$$

In this formula, W_i is the ability weight factor for every dimension, and $\sum W_i = 1$. The following dimension-matching algorithm shows how to calculate comprehensive dimension similarity.

In Algorithm 3, Line 1 compares the dimension similarity of a given event mention pair. Lines 2 to 5 compute the similarity of four dimensions as different similarity evidences, and Line 6 uses extended evidence theory to compute the comprehensive similarity of the event mention. The EDCoAGENES cluster method uses extended evidence theory to allocate dynamic weight and this method is more accurate than static weight. We will compare EDCoAGENES with other cluster methods in the experiment.

Algorithm 3 Dimension Matching

Input: Event mention pair $< em_i, em_j >$
Output: Event mention similarity $Sim(em_i, em_j)$

1. for every dimension pair $< d_i, d_j >$ in $< em_i, em_j >$
2. $Sim_{dimT} = dimTimeSimi(d_{i1}, d_{j1})$;
3. $Sim_{dimS} = dimSubjectSimi(d_{i2}, d_{j2})$;
4. $Sim_{dimO} = dimObjectSimi(d_{i3}, d_{j3})$;
5. $Sim_{dimA} = dimActivitySimi(d_{i4}, d_{j4})$;
6. $Sim(em_i, em_j) = computeSimi\ (Sim_{dimT}, Sim_{dimS}, Sim_{dimO}, Sim_{dimA})$;
   /*extended evidence theory add weight factor to every dimension similarity */
7. end for
8. return $Sim(em_i, em_j)$

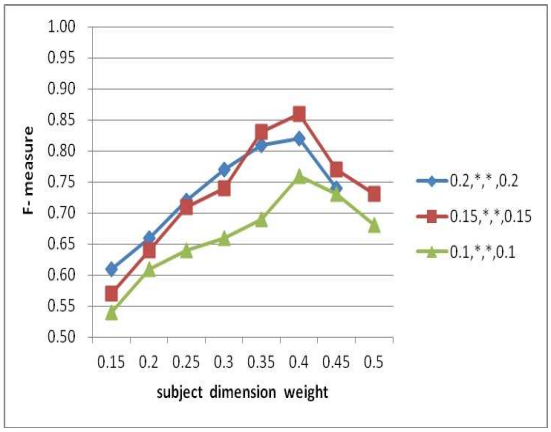

Fig. 6: Assessment of dynamic weight on four dimensions

5 Experiment

5.1 Experiment dataset

We extracted 9,000 event mentions in the food safety, phone, and computer fields as three experiment datasets. These event mentions were extracted from reports in news webpages during one week (from Aug 8, 2012 to Aug 15, 2012). We labeled the event mention that contains target entities and used eight dimensions to represent the event mention.

5.2 Experiment evaluation

The event detection discovered event clusters that contained many co-reference event mentions. To evaluate efficiency of the clustering method, we used an information retrieval evaluation method and divided the cluster results into four sets.

A = True Positives (event mentions that are clustered in a cluster is correct)

B = False Negatives (event mentions that are not clustered in a cluster is incorrect)

C = False Positives (event mentions that are clustered in a cluster is incorrect)

D = True Negatives (event mentions that are not clustered in a cluster is correct)

The precision, recall, and F-measure are calculated using the Equations (9), (10) and (11).

$$Recall = \frac{|A|}{|A| + |B|} \tag{9}$$

$$Precission = \frac{|A|}{|A| + |C|} \tag{10}$$

$$F - measure = \frac{2 \times Recall \times Precission}{Recall + Precission} \tag{11}$$

"*Recall*" evaluates the cover degree of correct clustering, "*precision*" evaluates the soundness of the clustering, and "$F - measure$" is the comprehensive evaluation.

According to the comprehensive dimension matching and co-occurrence constraint proposed in this paper, we tested the effectiveness of event detection in the following experimental aspects: (1) assessment of the extended evidence theory model to allocate dynamic dimension weights for clusters; (2) use of the co-occurrence constraint to reduce running times for event detection in multiple webpages; (3) comparison of the effectiveness of different event detection methods.

5.3 Experimental result and analysis

1. We used the extended evidence theory model to allocate weights and acquire the best weight distribution.

We used the extended evidence theory model to allocate dimension weight and obtain different cluster results. Important dimension needs to be allocated with a larger weight to obtain better weight allocation.

Fig. 6 shows some allocation plans by experiments. First, we fixed the time and activity dimensions by allocating different weights and adjusting the rest weight on the subject and object dimensions. (0.2, $*, *$, 0.2) means that the weight of time and activity are both fixed to 0.2, and the remaining 0.6 weight will be distributed to the subject and object dimensions. Fig. 6 shows that (0.15, 0.4, 0.3, 0.15) is the best weight allocation because the F-measure reaches the highest point. To compare the dynamic weight, we clustered the static weight (0.25, 0.25, 0.25, 0.25), and the F-measure is only 0.67.

2. Evaluation of the effect of co-occurrence constraint in restricting the cluster process for event detection in multiple webpage.

Fig. 7 shows the running time of two cluster methods on different event mention datasets. No noticeable difference was observed in the running time in small datasets, such as those that contain only 100 and 200

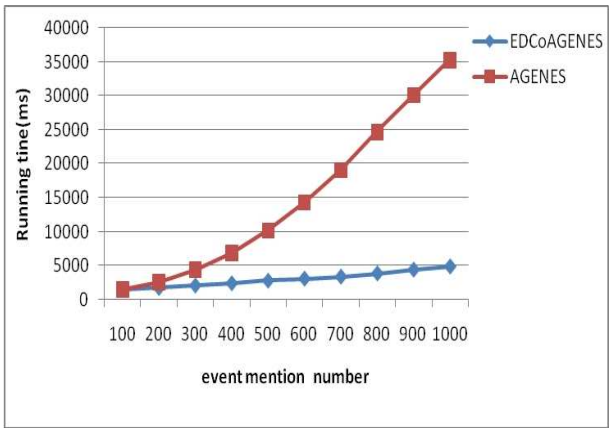

Fig. 7: Comparsion of running times of AGENES and EDCoAGENES

method	event mention amount			
	500	1000	1500	2000
	compare times			
AGENES	124750	499500	1124250	1999000
EDCoAGENES	17385 (12250+5135)	36735 (24500+12235)	57685 (36750+20935)	80235 (49000+31235)

Fig. 8: Compare match times with AGENES and EDCoAGENES

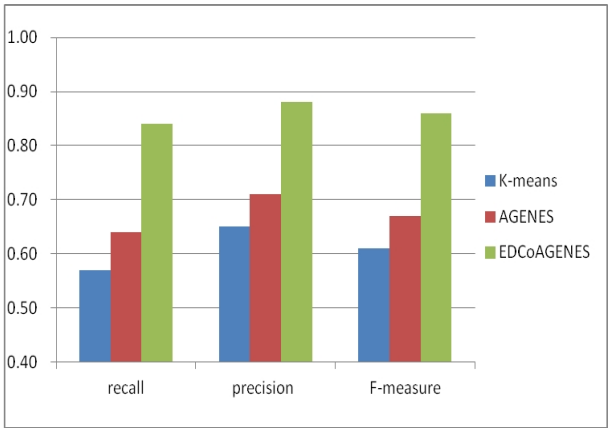

Fig. 9: Recall, precision, and F-measure of three methods on food safety dataset

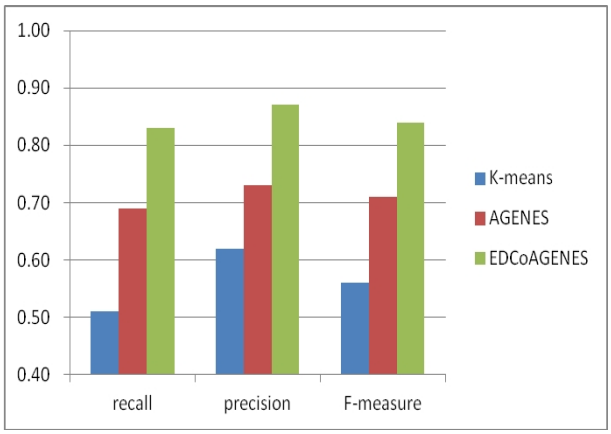

Fig. 10: Recall, precision and F-measure of three methods on phone dataset

event mentions. In larger datasets, EDCoAGENES uses a co-occurrence constraint that can significantly reduce running time. We propose the EDCoAGENES method for event detection in multiple webpages to reduce event match times and quantity of candidate matches events. Fig. 8 presents the different match times of the two cluster methods.

We chose four different event mention datasets to compare match times. In the EDCoAGENES line, (12,250 + 5,135) denotes that match times in a single webpage are added with match times between web pages, and 17,385 denote the sum of two parts. Fig.8 shows that EDCoAGENES, which uses two-stage event detection and a co-occurrence constraint, can reduce match times significantly.

3. We evaluated the effectiveness of using different cluster methods for event detection in multiple web pages. We compared the K-means, AGENES, and EDCoAGENES methods to detect events on food safety and phone datasets. The recall, precision, and F-measure of event detection are shown in Fig.9 and Fig.10.

We find that the EDCoAGENES method, which uses extended evidence theory to combine dimension-matching results and allocate dynamic dimension weights, has the higher recall, precision, and F-measure than other cluster methods (Figs. 9 and 10). To prove that this cluster method is not restricted to a special field, the experiment was conducted by using a different field dataset. The results show that EDCoAGENES is better than K-means or AGENES.

6 Conclusion

In this paper, we have proposed a method for detecting events in multiple webpages, namely, EDCoAGENES, which is based on comprehensive dimension matching and co-occurrence constraint theory. We detected single webpage events and found co-occurrence events in every single webpage. We presented an event co-occurrence constraint based on single webpage detection to reduce event match times and quantity of candidate matches events in multiple webpages. In event clustering process, we used eight dimensions to represent event mentions and compare the similarity of key dimensions. We used extended evidence theory to allocate the dynamic weight for different dimensions and combined the comprehensive

results as the similarity of event mentions. The experiment results demonstrate that this method can quickly detect various events in multiple webpages and can effectively reduce the number of co-reference events.

Acknowledgement

This work is supported by the National Key Technologies R&D Program (*No*.2012*BAH*54*F*01), Shandong Province Independent Innovation Major Special Project (*No*.2013*CXC*30201), the Natural Science Foundation of China (*No*.61303005) and the Shandong Distinguished Middle-aged and Young Scientist Encouragement and Reward Foundation (*No*.*BS*2012*DX*015).

References

[1] C. Y. Zhang, X. G. Hong, Z. H. Peng, Journal of Software, **23**, 2612-2627 (2012).
[2] C. J. Fillmore, Quademi di Semantica, 222-253, (1986).
[3] Bo Zhang and Zhicai Juan, Modeling User Equilibrium and the Day-to-day Traffic Evolution based on Cumulative Prospect Theory, Information Science Letters, **2**, 9-12 (2013).
[4] D. S. Anish, J. Alpa and C. Yu, Proc of the 4th International Conference on Web Search and Web Data Mining, Hong Kong, China, 207-216 (2011).
[5] J. Yao, B. Cui, Y. Huang and Y. Zhou, World Wide Web, **15**, 171-195 (2012).
[6] W. S. Pan, S. Z. Chen, Z. Y Feng, Applied Mathematics & Information Sciences, **7**, 675-681 (2013).
[7] M. K. Agarwal, K. Ramamritham and M. Bhide, Proceedings of the VLDB Endowment, **5**, 980-991 (2012).
[8] R. Yan, Y. Li, Y. Zhang and X. Li, Information Retrieval Technology, 490-501, (2010).
[9] D. Shan, W. X. Zhao, R. Chen, B. Shu, Z. Wang, J. Yao, H. Yan and X. Li, Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **18**, 1564-1567 (2012).
[10] W. X. Zhao, R. Chen, K. Fan, H. Yan, and X. Li, Proceeding of the 50th Annual Meeting of the Association for Computational Linguistics Jeju, **50**, 43-47 (2012).
[11] M. Naughton, N. Stokes and J. Carthy, Information retrieval, **13**, 132-156 (2010).
[12] B. Jiang, M.-x. Zhu, and J.-l. Wang, Journal of Computers, **8**, 85-90 (2013).
[13] D. A. Smith, Proceeding of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, **25**, 73-80 (2002).
[14] Z. Lei, J. Liao, D. Li and L. Wu, Proceedings of the 4th International Conference on Intelligent Computing, **4**, 872-879 (2008).
[15] T. Xiëski, A. Nowak-Brzeziska and A. Wakulicz-Deja, Proceeding of the 8th International Conference, **8**, 142-149 (1956).
[16] L. Zheng, L. Li, W. Hong and T. Li, Expert Systems with Applications, **40**, 2127-2136 (2013).
[17] Z. M. Zhong and Z. T. Liu, Pattern Recognition and Artificial Intelligence, **23**, 307-313 (2010).
[18] C. Li, Z. Xu and T. Luo, Pattern Recognition Letters, **34**, 155-162 (2013).
[19] S. Salicone, IEEE Instrumentation & Measurement Magazine, **16**, 18-23 (2013).
[20] http://www.keenage.com/.
[21] P. S. Medeiros Dos Santos and G. H. Travassos, Electronic Notes in Theoretical Computer Science (ENTCS), **292**, 95-118 (2013).

Yuanzi Xu received the Bachelor's degree from Zhengzhou University in 2006, and the Master's degree from Shandong University in 2009. She is currently a Ph.D candidate in Shandong University of computer science and technology. Her research interests are in the areas of Web information integration and event detection.

Qingzhong Li received the Master's degree from Shandong University in 1989, and the Ph.D from the Institute of Computing Technology Chinese Academy of Sciences in 2000. He is currently a professor in Shandong University. His research interests are in the areas of Web information integration and large-scale network data management.

Zhongmin Yan received the Ph.D degree from Shandong University in 2010. She is currently an associate professor in Shandong University. Her research interests are in the areas of Web information integration and Web data management.

Wei Wang received the Bachelor's degree from Shandong University in 2012. He is currently a master candidate in Shandong University. His research interests are Web information integration and event detection.