*i*WED: An Integrated Multigraph Cut-based Approach for Detecting Events from A Website

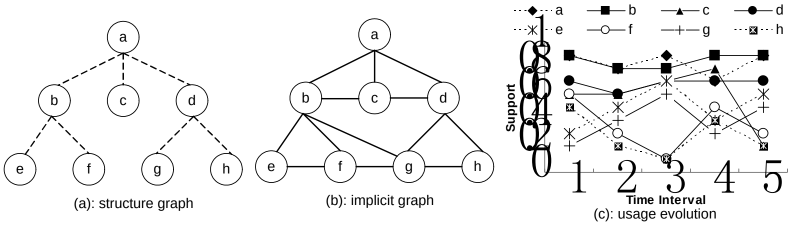
Qiankun Zhao   Sourav S Bhowmick   Aixin Sun

CAIS, Nanyang Technological University, Singapore
qkzhao@pmail.ntu.edu.sg, assourav@ntu.edu.sg, axsun@ntu.edu.sg

Abstract. The web is a sensor of the real world. Often, content of web pages correspond to real world objects or events whereas the web usage data reflect users' opinions and actions to the corresponding events. Moreover, the *evolution patterns* of the web usage data may reflect the evolution of the corresponding events over time. In this paper, we present two variants of *i*WED(**I**ntegrated **W**eb **E**vent **D**etector) algorithms to extract events from website data by integrating *author-centric data* and *visitor-centric data*. We model the website related data as a *multigraph*, where each vertex represents a web page and each edge represent the *relationship* between the connected web pages in terms of *structure*, *semantic*, and *usage pattern*. Then, the problem of event detection is to extract *strongly connected subgraphs* from the multigraph to represent real world events. We solve this problem by adopting the normalized graph cut algorithm. Experiments show that the *usage pattern*s play an important role in *i*WED algorithms and can produce high quality results.

1 Introduction

The web has invaded our lives. In some sense, the web is a sensor of the real world. Specifically, it has been observed that events and objects are often represented by a set of web pages but not as individual web pages [4, 8]. Consequently, a large body of literature has focused on extracting real world events or objects from web data [4, 5, 8, 11, 12]. These approaches can be classified into two groups: *structure-based* extraction and *content-based* extraction. In the structure-based approaches, the website structures, hyperlink structures, and URLs are used to extract sets of web pages corresponding to events and objects [8, 4]. In the content-based extraction, content of web pages are segmented and categorized into subgroups that correspond to different topics, events, and stories using techniques such as natural language processing and probability models [1, 11, 12]. At the same time, such extraction results have been proved useful in many applications such as organizing the website structure [8], restructuring the web search results [4], terrorism event detection [9], and *Photo Story* and *Chronicle* [5].



(a): structure graph

(b): implicit graph

(c): usage evolution

Fig. 1. Web data representation

Data associated with a set of web pages in a web site can be classified into two types: *author-centric* and *visitor-centric*. *Author-centric* data refers to a set of hyperlinked web pages that describes certain object or event, while *visitor-centric* data refers to the web access sequences of these pages and describes how the web pages are accessed in the history. Observe that author-centric data describes authors' point of view while visitor-centric data reflect the web visitors' point of view.

We observed that existing event and object extraction approaches only analyze the author-centric data. *These techniques ignore visitor-centric data.* However, often it may not be possible to distinguish different events related to the same topic by using the author-centric data only. This is because events belonging to the same topic often share a set of keywords and the pages containing these different events often are connected by hyperlinks. For example, web pages talking about different *car accidents* tend to share keywords like *car*, *accidents*, and *crash*. Also, these pages may be connected as they belong to the same topic (car accident). Hence, it is difficult to distinguish one car accident from another based on only keywords and hyperlink structure.

In this paper, we consider visitor-centric data along with author-centric data to detect real-world events. In other words, we integrate visitor-centric and author-centric data to distinguish different events under the same topic. The major differences between our event detection approach and the related research [1, 11, 12, 5, 4, 2] are twofold. First, all the above works focus on either the author-centric or the visitor-centric data, while our approach incorporates the visitor-centric data along with the author-centric data. Second, the temporal property of the visitor-centric data is utilized in our approach to improve the event detection accuracy.

For example, suppose Figure 1(a) shows a subset of hyperlinked web pages; Figure 1(b) shows the *implicit links* extracted from the corresponding usage data; and Figure 1(c) shows the *evolution pattern* of web usage data (the y-axis shows the frequency of a web page being accessed over the time intervals shown in the x-axis). Here, there is an *implicit link* between two web pages if and only if they were accessed consecutively in the web access sequences [10]. The *evolution pattern* of web usage data refers to how the web pages changed in the history in terms of their supports [13].

It can be observed that from only Figure 1(a), it is difficult to distinguish sibling pages such as e and f even if they correspond to different events. However, with the evolution of web usage data as shown in Figure 1(c), connected web pages with similar content but corresponding to different events can be distinguished. For example, in Figure 1(c), pages e and g have similar evolution pattern while pages e and f have different evolution pattern. At the same, web pages that are not connected by hyperlinks but corresponding to the same event can be identified using implicit links in Figure 1(b), since they are expected to be accessed together. As shown in Figure 1(b), the implicit link between web pages b and g, which are not connected by hyperlink in Figure 1(a), implies that b and g have a possibility to represent the same event. In this paper, we focus on detecting events in a specific website as it is extremely difficult to gather web usage data of the entire web. The contributions of this paper are as follows.

- To the best of our knowledge, this is the first approach that detects website level events by integrating web structure, web content, and web usage data and its evolution patterns.
- A *multigraph* is proposed to model website related data in terms of structure, semantics, and usage patterns by integrating the *author-centric* and *visitor-centric* data.
- We present two variants of *i*WED algorithms, called *fusion-base graph cut* and *level-wise graph cut*, to detect events from the multigraph. These algorithms are inspired by the normalized graph cut algorithm widely used in image and video object extraction [6]. Experiment results show that the *i*WED event detection algorithms can produce high quality results.

2 Website Data Representation and Problem Statement

In this section, we first discuss how to represent web structure, web content, and web usage data of a web site using *structure graph*, *content graph*, and *usage graph*, respectively. Then, we present how these three types of graphs are integrated using a *multigraph*, followed by the problem statement of website-based event detection.

2.1 Structure Graph

The web structure data here refers to the set of web pages and hyperlinks between them. It can be modelled as a *structure graph*, $G_s = \langle V_s, E_s \rangle$, where each vertex in V_s is a web page and each edge in E_s represents the *structure similarity* (will be defined later) between the two pages that are connected by this edge. Note that the *structure similarity* is defined to reflect the similarity between web pages in terms of structure. The intuition is "two web pages are structurally *similar* if they are linked with *similar* web pages" [3]. As the base case, we consider a web page maximally similar to itself, to which we can assign a structure similarity score of *1*. With this intuition, given two web pages i and j in V_s, the *structure similarity* is defined as:

$$S_s(i,j) = \frac{C}{|D(i)| * |D(j)|} \sum_{m=1}^{|D(i)|} \sum_{n=1}^{|D(j)|} S_s(D_m(i), D_n(j))$$

Here C is a constant between 0 and 1, $|D(i)|$ is the degree of vertex i in the graph and $D_m(i)$ is the m^{th} neighbor of vertex *i*. It is obvious that this similarity is an iterative function where similarities between web pages are propagated through recursions. That is, the value of $S_s(i,j)$ in the t^{th} iteration, denoted as S_{s_t}, is based on the values of the t-1^{th} iteration. More over it has been proved that this recursive function is nondecreasing and it will converge eventually [3]. We initialize the recursions with S_{s_0}: if *i*=*j*, then $S_{s_0}(i,j)$=1; otherwise $S_{s_0}(i,j)$=0.

2.2 Content Graph

The web content data refers to the content of each web page. The web content data is modelled as a *content graph*, $G_c = \langle V_c, E_c \rangle$, where each vertex in V_c is a web page and each edge in E_c represents the *semantic similarity* between two pages. It has been experimentally proven that *cosine measure* is one of the best measures for web content clustering [7]. Hence, we use the cosine measure to quantify *semantic similarity* between two pages. Given a web page i, using some stemming algorithm, it will be represented as a vector, $\overrightarrow{X_i}$, which correspond to the *TF.IDF* of the keywords after stemming [7]. Then, the *semantic similarity* between two web pages i and *j*, denoted as $S_c(i,j)$, is defined as:

$$S_c(i,j) = \frac{(\overrightarrow{X_i} \bullet \overrightarrow{X_j})}{||\overrightarrow{X_i}|| \cdot ||\overrightarrow{X_j}||}$$

where $(\overrightarrow{X_i} \bullet \overrightarrow{X_j})$ is the dot product of the two vectors and $||\overrightarrow{X_j}||$ denote the length of vector $\overrightarrow{X_j}$.

2.3 Usage Graph

The usage data refers to the access log of the web pages. It also can be modelled as a graph, called *usage graph*, $G_u = \langle V_u, E_u \rangle$, where each vertex in V_u is a web page and each edge in E_u represents the *usage pattern-based similarity* between two pages. Firstly, we review some of the literature in web usage mining.

In general, web usage data record the interactions between web users and the web server. A web access sequence ($\mathcal{WAS}$) is an ordered list of pages accessed by a user, i.e., $A = \langle (p_1, t_1), (p_2, t_2), \ldots, (p_n, t_n) \rangle$, where p_i is a web page, t_i is the time when p_i was accessed and $t_i \leq t_{i+1}$ $\forall$ $i = 1, 2, 3, \ldots, n-1$. Similar to [13], the $\mathcal{WAS}$s can be represented as a sequences of $\mathcal{WAS}$ *group* based on the user-defined time interval. A $\mathcal{WAS}$ *group* (denoted as G) is a bag of $\mathcal{WAS}$s that occurred during a specific time period. Let t_s and t_e be the start and end times of a period. Then, $G = [A_1,\ A_2,\ \ldots,\ A_k]$ where p_i is included in $\mathcal{WAS}$ A_j for $1 < j \leq k$ and p_i was visited between t_e and t_s. As a result, the historical web log data is divided into a sequence of $\mathcal{WAS}$ groups. Let $H_G = \langle\ G_1,\ G_2,\ G_3,\ \ldots,\ G_k\ \rangle$ be a sequence of k $\mathcal{WAS}$ groups generated from the historical web log data. Given a web page i, let $H_i = \langle\ \Phi_1(i),\ \Phi_2(i),\ \Phi_3(i),\ \ldots,\ \Phi_k(i)\ \rangle$ be the sequence of support values of i in H_G. Note that, for $1 \leq t \leq k$, $\Phi_t(i) = \frac{\mathcal{N}}{|G_t|}$, where $\mathcal{N}$ is the number of $\mathcal{WAS}$s that contain i.

Given two web pages, i and j, with the corresponding web usage data, the *usage pattern-based similarity*, denoted as $S_u(i, j)$, is defined as:

$$S_u(i,j) = \lambda \times e^{-D} + (1-\lambda) \times \frac{\sum_{t=1}^{k}(\Phi_t(\langle i,j \rangle) + \Phi_t(\langle j,i \rangle))}{\sum_{t=1}^{k}(\Phi_t(i) \cup \Phi_t(j))},$$

where $D = \sqrt{\sum_{t=1}^{k} |\Phi_t(i) - \Phi_t(j)|^2}$.

Note that, the usage pattern-based similarity is a linear combination of the *evolution pattern-based similarity* and the *implicit link-based similarity.* The evolution pattern-based similarity is denoted as e^{-D}, where D is the *Euclidian distance* between the support sequences *H(i)* and *H(j).* The implicit link-based similarity is represented as the percentage of $\mathcal{WAS}$s that contain i and j consecutively against the total number of $\mathcal{WAS}$s that contain at least one of i and j . Here, λ and $1 - \lambda$ are the weights of evolution pattern-based similarity and the implicit link-based similarity. It is obvious that both the evolution pattern-based similarity and implicit link-based similarity are within the range between 0 and 1. Similarly, the usage pattern-based similarity is between 0 and 1.

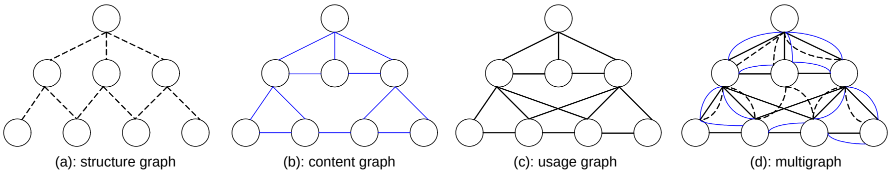


Fig. 2. Web data representation

2.4 Multigraph

We merge the above three graphs using a *multigraph*, which includes web structure, web content, and web usage data in a website. A *multigraph* is a graph whose edges are unordered pairs of vertices, and the same pair of vertexes can be connected by multiple edges. In this case, there are three edges for each pair of vertexes. These three edges represent the edges of structure graph, content graph, and usage graph.

Definition 1. *[**Multigraph**] A multigraph is represented as a 3-tuple* $M = \langle\ V,\ E,\ f\ \rangle$, *where* V *is a set vertexes ,* E *a set of edges, and* f *is a function* $f\ (e_i) = \{\{u,v\}|u, v \in V;\ u \neq\ v\ \}$ *that takes an edge* $e_i \in E$ *and returns the set of web pages* u *and* v *that are connected by* e_i*. Two edges* e_i *and* e_j *are called parallel or multiple edges if* $f\ (e_i) = f\ (e_j)$*.*

An example of the multigraph representation of website data is shown in Figure 2 with the corresponding structure graph, content graph, and usage graph. Note that, the similarities between disconnected web pages are *0* and the weights of the edges represent the corresponding similarity values.

Website-based Event Detection Problem: Based on the multigraph representation of the website related data, each real world event corresponds to a strongly connected subgraph in the multigraph. That is, a real world event can be represented as a set of structurally and semantically strongly connected web pages with similar usage patterns in the multigraph. The website based event detection problem is to extract such subgraphs from the multigraph representation.

3 *i*WED Algorithms

In this section, we present the *i*WED event detection algorithms based on the multigraph representation of the website data. To extract the strongly connected subgraphs from a graph, different graph cut algorithms have been proposed. In this paper, we adopt the normalized graph cut algorithm, which is widely used in object extraction from image data and frame segmentation of video data [6].

The three similarity measures, S_s, S_c, and S_u, introduced in Section 2 can be classified into two categories: *topic similarity* and *evolution similarity*. Topic similarity is the combination of the structure similarity (S_s) and the semantic similarity (S_c), while evolution similarity is the usage pattern-based similarity (S_u). Based on these two categories, we propose two variants of *i*WED algorithms for cutting the multigraph. The first approach, called the *fusion approach*, fuses the two types of similarity measures together and cut the graph by treating the multiedges between two vertexes as a single edge. The second approach, called the *level-wise approach*, cuts the graph with the topic and evolution similarity measure separately. We now elaborate on these two approaches.

Fusion Approach: The fusion approach, denoted as FUS, integrates the three similarity measures together using linear combination with different weights. Such kind of fusion has been extensively used in combining different types of similarity measures in web content analysis [3]. In the fusion approach, a new similarity $\mathcal{S}$ is proposed as: $\mathcal{S} = \alpha S_s + \beta S_c + \gamma S_u$, where α, β, γ are the weights for the corresponding similarity measure, and $\alpha+\beta+\gamma=1$. Then, the multigraph is transformed to a normal graph, where the weight of each edge is represented by $\mathcal{S}$. The graph is then cut using the normalized graph cut algorithm.

Level-wise Approach: In the level-wise approach, the topic similarity and the evolution similarity are used to cut the multigraph separately. Note that, the topic similarity, denoted as S^T, defined as the fusion of structure similarity and semantic similarity. There are two alternative level-wise approaches. In the first approach, denoted as LTF (Level-wise Topic First), the multigraph is cut based on the topic similarity, which corresponds to only two types of edges in the multigraph, and the result, C_T, is returned. Then, each subgraph in C_T is cut again based on the evolution similarity and the final result, C_F, is returned. In the second approach, denoted as LEF (Level-wise Evolution First), the multigraph is first cut based on the evolution similarity and the result, C_E, is returned. Then, each subgraph in C_E is cut again using the topic similarity and the result C_F is returned. The underlying intuition is that, in the first approach, web pages are clustered into semantic topics before they are clustered into events as each event is expected to be a set of semantically similar web pages that have similar usage patterns. In the second approach, firstly web pages that correspond to similar types of events are gathered together and then clustered based on their semantic relationships.

For both the fusion approach and the level-wise approach, we present the clustering results with a hierarchical structure. That is, at the first recursion of the 2-way graph cut algorithm, there are two partitions. After that each partition is further cut into two child partitions and so on. However, not all the subgraphs correspond to real world events. To identify real world events and exclude outliers, we propose an *intra-cluster similarity measure*, $\mathcal{S}_{intra}(G')$, for any subgraph G':

$$\mathcal{S}_{intra}(G') = \frac{2\sum_{i}^{|G'|}\mathcal{S}(i,j)}{|G'| \times (|G'|-1)}$$

, where $i \neq j$ and $i, j \in G'$. Based on this similarity measure, a threshold τ in the range of [0, 1], is proposed to distinguish the event-based subgraph and the non-event-based subgraph. A subgraph, G' in the cut results corresponds to a real world event if and only if $\mathcal{S}_{intra}(G') \geq \tau$.

4 Performance Evaluation

In this section, the experimental results are presented to show the performance of our proposed event detection approaches. The three approaches, FUS, LTF, and LEF, are implemented and compared to the *baseline* approach, Bl, which only takes the structure and content of web pages using the corresponding similarity measures proposed in Section 2.

In our experiments, a synthetic e-commerce website dataset is used. Even though there are some real web usage datasets available, but due to privacy issue the original URLs and web pages are not available and cannot be used in our experiments. The synthetic dataset we generated consists of 300 products and 2000 unique web pages. The 300 products belong to 5 categories, where the content of the web pages are generated according the attributes of products in different categories (we use the schema extracted from http://www.bargaincity.com.sg, which is the one of the biggest e-commerce websites in Singapore). The usage data are generated in three steps. Firstly, the web access sequences are generated using uniform random generation. Then, we synthesized a list of 100 events (20 burst events such as one day only promotion and release of new products, 40 periodic events such as weekend promotion and new semester promotion, 20 increasing events such as price of a popular product keeps decreasing, 20 decreasing events such as some products are fading out of the market). Lastly, some noise access sequences are randomly inserted into the web usage data to mimic the real life usage data. In total, there are 10,000,000 unique page request in the synthetic web usage data, which are partitioned into 100 access groups.

4.1 Evaluation Measures

As the event detection results are set of events, which consist of sets of web pages, it is different from existing classification algorithms. Although, we have the set of labelled events with corresponding web pages, the precision and recall measures in our event detection approach are different for the following reasons. Since an event consists of many web pages, the event may be detected but the corresponding web pages may not be accurate. That is, some pages may be missed and some non-related pages may be included. For example, given a real world event $E = \{P_1, P_2, P_3, P_4, P_5\}$, there may be one corresponding event $E' = \{P_1, P_3, P_4, P_7, P_8\}$ in the detection results. Moreover, for one real world event, there may be more than two corresponding events in the results. For example, given a real world event $E = \{P_1, P_2, P_3, P_4, P_5\}$, there may be two corresponding events $E' = \{P_1, P_3, P_4, P_7, P_8\}$ and $E'' = \{P_2, P_5, P_9\}$ in the detection results. We propose precision/recall measure for event detection based on the commonly-used precision/recall from IR.

Let $\mathcal{E} = \{E_1, E_2, \cdots, E_n\}$ be the set of detected events based on our proposed approach and $\mathcal{E}' = \{E'_1, E'_2, \cdots, E'_m\}$ be the set of labelled events in the dataset, where each event E_i consists of a set of web pages $\{P_{i1}, P_{i2}, \cdots, P_{ik}\}$. For each E_i, the corresponding real event E'_j with the largest value of $|E_i \cap E'_j|$ is selected, $|E_i|$ is the number of pages included in that event while $|E_i \cap E'_j|$ is the number of common pages included in both E_i and E'_j. Also, for each real world event E'_j, the corresponding event E_i with the largest value of $|E_i \cap E'_j|$ is selected from the results. Moreover, for different events in the real world, their corresponding events in the results should be different and vise versa. Then, the precision and recall are defined as:

$$. \qquad Pr = \frac{\sum_i^{|\mathcal{E}|} \frac{|E_i \cap E'_j|}{|E_i|}}{|\mathcal{E}|} \quad Re = \frac{\sum_j^{|\mathcal{E}'|} \frac{|E_i \cap E'_j|}{|E'_j|}}{|\mathcal{E}'|}$$

4.2 Experimental Results

Two sets of experiments have been conducted to evaluate our proposed event detection approaches. Firstly, comparison of our proposed event detection approaches with the baseline approach is presented. Secondly, we show the effects of intra-similarity threshold τ on the quality of the detected events. Within each set of results, both the overall performance and the performance for each type of events are presented. Lastly, we discuss about how to set the fusion parameters in the *FUS* approach.

(a) All events

Alg	Pr	Re	F_1
Bl	0.376	0.108	0.168
FUS	**0.729**	**0.696**	**0.712**
LTF	0.591	0.412	0.486
LEF	0.684	0.625	0.653

(b) Burst events

Alg	Pr	Re	F_1
Bl	0.531	0.192	0.282
FUS	**0.892**	**0.751**	**0.815**
LTF	0.674	0.582	0.625
LEF	0.873	0.749	0.806

(c) Periodic events

Alg	Pr	Re	F_1
Bl	0.227	0.098	0.137
FUS	**0.678**	**0.622**	**0.649**
LTF	0.535	0.491	0.512
LEF	0.647	0.562	0.602

(d) In/Decreasing events

Alg	Pr	Re	F_1
Bl	0.483	0.298	0.364
FUS	**0.912**	**0.895**	**0.904**
LTF	0.692	0.769	0.728
LEF	0.875	0.864	0.869

(e) FUS

τ	Pr	Re	F_1
0.1	0.314	0.452	0.371
0.3	0.729	0.696	0.712
0.5	0.758	**0.712**	0.734
0.7	**0.841**	0.709	**0.769**
0.9	0.413	0.422	0.417

(f) LEF

τ	Pr	Re	F_1
0.1	0.279	0.354	0.312
0.3	0.591	0.412	0.486
0.5	0.681	0.527	0.594
0.7	**0.748**	**0.699**	**0.723**
0.9	0.324	0.435	0.371

Table 1. Event Detection Results

Note that, the λ value in the usage pattern-based similarity is set to 0.5 for the following experiments.

Table 1(a) shows the performance of the four approaches with the precision, recall, and F_1 measure[1]. It can be observed that the LEF, FUS, and LTF approaches outperform the baseline approach, *Bl*, which shows the improvement of integrating the usage data and their evolution patterns. Among our proposed approaches, the LEF and FUS archive better performances than the LTF approach. However, the FUS and LEF approaches can discover them as a single event. This is because some of the synthetic events in our dataset usually cover more than one semantic topic. Tables 1(b), (c), and (d) show the performance of our approaches with respect to different types of events.

In the above experiments, weights of the three similarity measures are set to *0.31*, *0.20*, and *0.49*, which are experimentally proved to be the optimal values for our dataset. The threshold for intra-cluster similarity is set to *0.6*. Tables 1(e) and (f) show the quality of the event detection results of the FUS and LEF approaches by varying the corresponding τ values. The results are for all types of events. Observe that the effects of threshold τ are similar for the three types of events. When the value of τ increases from *0.3* to *0.7*, the quality of the event detection results becomes better; when the value of τ increases from *0.7* to *0.9*, the quality of the event detection results becomes worse. This is because when the threshold for intra-cluster similarity is too small/large, the number of events detected may be too many/few. While the number of real world

[1] The F_1 measure is computed as $F_1 = \frac{2*Pr*Re}{Pr+Re}$

event is fixed, the performance of the approaches decreases when the threshold is close to the two extremes.

From the results shown in Table 1, it is evident that the *FUS* approach performs relatively better than other approaches in most cases. This is because, in the *FUS* approach, the weights of different types of similarities can be tuned. In our experiments, we show the average results of the *FUS* approach. It can be observed that the usage pattern-based similarity significantly improves the clustering results. Moreover, we observed that the structure similarity is less important than the usage pattern-based similarity but more important than the content similarity.

5 Conclusions

This work is motivated by the fact that existing event and object detection approaches only analyze the content and structure data of a website. In this paper, we integrate the author-centric and visitor-centric data to detect real-world events. Experimental results show that our proposed approaches can produce promising results.

References

1. J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *SIGIR*, 314–321, 2003.
2. S. Gunduz and M. T. Ozsu. A web page prediction model based on click-stream tree representation of user behavior. In *SIGKDD*, 535–540, 2003.
3. G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *SIGKDD*, 538–543, 2002.
4. W.-S. Li, K. S. Candan, Q. Vu, and D. Agrawal. Retrieving and organizing web pages by "information unit". In *WWW*, 230–244, 2001.
5. Z. Li, M. Li, and B. Wang. Probabilistic model of retrospective news event detection. In *SIGIR*, 2005.
6. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans PAMI*, 22(8):888–905, 2000.
7. A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *AAAI Workshop of AI for Web Search*, pages 58–64, 2000.
8. A. Sun and E.-P. Lim. Web unit mining: finding and classifying subgraphs of web pages. In *CIKM*, 108–115, 2003.
9. Z. Sun, E.-P Lim, K. Chang, T.-K. Ong and R. K. Gunaratna Event-Driven Document Selection for Terrorism Information Extraction. In *IEEE ISI*, 37–48, 2005.
10. G.-R. Xue, H.-J. Zeng, Z. Chen, W.-Y. Ma, H.-J. Zhang and C.-J. Lu. Implicit link analysis for small web search. In *SIGIR*, 56–63, 2003.
11. Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In *SIGKDD*, 688–693, 2002.
12. J. Zhang, Z. Ghahramani, and Y. Yang. A probabilistic model for online document clustering with application to novelty detection. In *NIPS*, 1617–1624. 2005.
13. Q. Zhao and S. S. Bhowmick and L. Gruenwald. WAM-Miner: In the Search of Web Access Motifs from Historical Web Log Data. In *Proceedings of CIKM* 2005.