Discovering Event Evolution Graphs from Newswires

Christopher C. Yang
Department of System Engineering and
Engineering Management
The Chinese University of Hong Kong
yang@se.cuhk.edu.hk

Xiaodong Shi
Department of System Engineering and
Engineering Management
The Chinese University of Hong Kong
xdshi@se.cuhk.edu.hk

ABSTRACT

In this paper, we propose an approach to automatically mine event evolution graphs from newswires on the Web. Event evolution graph is a directed graph in which the vertices and edges denote news events and the evolutions between events respectively, in a news affair. Our model utilizes the content similarity between events and incorporates temporal proximity and document distributional proximity as decaying functions. Our approach is effective in presenting the inside developments of news affairs along the timeline, which can facilitate users' information browsing tasks.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: information filtering, retrieval models, clustering

General Terms

Algorithms, Experimentation, Measurement

Keywords

Web content mining, event evolution, event evolution graph, knowledge management

1. INTRODUCTION

With a large volume of digitized news information published on the Web, it becomes increasingly difficult and time consuming for human users to conceptualize a specific interesting news affair. Two existing approaches, searching and clustering, address this problem from different aspects. Searching retrieves a list of relevant news stories to the topic and clustering organizes discovered news stories into a flat hierarchical structure, either manually or automatically. Both approaches fail to present the inside structure of the evolutions of news events within the news affair. Some recent work [1, 2] made good attempts in modeling this type of evolution structures. In this work, we propose to use event evolution graph to model the evolutions of news events within a news affair.

2. EVENT EVOLUTION GRAPH

Users are often as interested in the development of news events as in their details when reading news stories from newswires (such as CNN or BBC). However, existing techniques mostly deals with the retrieving of relevant documents to a given topic and how to organize them into a hierarchical structure, as in Topic Detection and Tracking (TDT). Our approach not only identifies the news events from news stories but also models the developments or evolutions between these news events.

We define **event evolution** as the directional dependencies or relatedness, which exhibit the track of event development, between two events inside a same news affair. For two events A and B, if the event evolution from event A to B exists, we assert that there is an **event evolution relationship** between event A and B. For example, considering the following two events "Columbia space shuttle crashed" (*A*) and "NASA conducted investigations into the shuttle crash" (*B*), we can claim there is an event evolution relationship between event A and B.

Assume there are totally n events $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ in a given news affair. Each event ε_i can be represented as $<t_i, S_i>$, where t_i is the timestamp of ε_i and S_i is the set of news stories $\{d_{i1}, d_{i2}, ..., d_{im}\}$ that belong to ε_i ($S_i \neq \emptyset$, and $\forall S_i, S_j, S_i \cap S_j = \emptyset$). t_i can be further represented as a time interval $[s_i, e_i]$; when $s_i = e_i$, the timestamp becomes a spot time. We denote the event evolution relationship from event ε_i to event ε_j as $(\varepsilon_i, \varepsilon_j)$ ($i \neq j$).

Given the set of events and the event evolution relationships between them, the best structure to present the blueprint of that news affair is a directed graph. Literally, **event evolution graph** is defined as a directed acyclic graph (DAG) $\mathbf{G} = (\mathbf{V}, \mathbf{L})$ consisting of events as the nodes $\mathbf{V}$ and event evolution relationships as the directed edges $\mathbf{L}$ between nodes, where $\mathbf{V} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ and $\mathbf{L} = \{(\varepsilon_i, \varepsilon_j)\}$ $(\varepsilon_i, \varepsilon_j \in \mathbf{V})$. Figure 1 displays a part of the sample event evolution graph for the news affair of Beslan school hostage tragedy that happened in Russia in 2004.

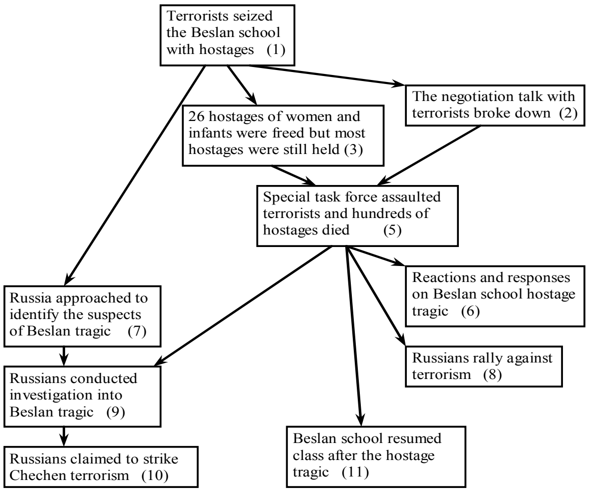
Terrorists seized the Beslan school with hostages (1)
The negotiation talk with terrorists broke down (2)
26 hostages of women and infants were freed but most hostages were still held (3)
Special task force assaulted terrorists and hundreds of hostages died (5)
Reactions and responses on Beslan school hostage tragic (6)
Russia approached to identify the suspects of Beslan tragic (7)
Russians rally against terrorism (8)
Russians conducted investigation into Beslan tragic (9)
Russians claimed to strike Chechen terrorism (10)
Beslan school resumed class after the hostage tragic (11)

Figure 1. The partial event evolution graph for "Beslan school hostage crisis". The numbers in the bracket indicates their temporal orderings.

3. MINING EVENT EVOLUTION GRAPHS

Makkonen [1] first developed the ontologies, including general terms, locations, names, and temporals, to measure the relatedness of events. It was later found by Nallapati et al. [2] that location and name features as suggested by Makkonen [1] were not effective in modeling event evolutions. In our work we utilize the event content similarity to measure the confidence of event evolution relationships. In addition to that, we incorporated two

WWW 2006, May 22–26, 2006, Edinburgh, UK.

decaying factors into the function, i.e. temporal proximity and document distributional proximity.

- Event Content Similarity

We measure the content similarity between events by the cosine similarity between their event term vectors. This coincides with our intuition: for two events between which there is an event evolution relationship, they often share similar key vocabularies in their news stories and reference one the other in their texts.

The event term vector of event ε_i is computed as the average of the document term vectors of stories that belong to ε_i. Note that here only term frequency is applied instead of the TF-IDF measure that is common in other information retrieval techniques.

The event content similarity between events ε_i and ε_j is:

$$cs(\varepsilon_i,\, \varepsilon_j) = cosine\_sim(etv(S_i),\, etv(S_j)) \qquad (1)$$

where $etv(S_i)$ and $etv(S_j)$ are the event term vectors of ε_i and ε_j.

- Temporal Proximity

We first define the *temporal distance* between two events ε_i and ε_j as (assuming $s_i \leq s_j$):

$$d(t_i, t_j) = \begin{cases} s_j - e_i & (if\ \ e_i \leq s_j) \\ 0 & (if\ \ e_i > s_j) \end{cases} \qquad (2)$$

The temporal distance between events is also helpful in measuring the confidence of event evolution relationships. Intuitively if two events are far away from each other along the timeline, then the event evolution is less likely to exist between them than those events close to each other. We use the *temporal proximity* between events to measure the relative temporal distance between two events, i.e. (assuming $s_i \leq s_j$):

$$tp(\varepsilon_i, \varepsilon_j) = e^{-\alpha\left[\frac{d(t_i,t_j)}{T}\right]} \qquad (3)$$

where T is the *event horizon* defined as the time-span of the entire news affair. α is the time decaying weight which is between 0 and 1.

- Document Distributional Proximity

Temporal proximity fails to perform well in some cases, such as when there is a burst of events and stories, usually at the beginning stage of a news affair. Therefore, we utilize the distribution of documents in order to counterwork the shortcomings of temporal proximity. We define the *document distributional proximity* as a second decaying function:

$$df(\varepsilon_i, \varepsilon_j) = e^{-\beta\frac{m}{N}} \qquad (4)$$

where m is the number of documents that belong to the events happening in-between event ε_i and ε_j. N is the total number of documents in the topic. β is a decaying factor.

Finally the confidence of event evolution relationships is:

$$conf\left((\varepsilon_i, \varepsilon_j)\right) = \begin{cases} 0 & if\ i{=}j\ or\ s_i > s_j \\ cs(\varepsilon_i, \varepsilon_j) \times tp(\varepsilon_i, \varepsilon_j) \times df(\varepsilon_i, \varepsilon_j) & if\ i \neq j\ and\ s_i \leq s_j \end{cases} \qquad (5)$$

To construct the event evolution graph, we simply assume that there is a hypothesized event evolution relationship between every pair of events in the news affair. We then compute the confidence of these event evolution relationships and filter away further undesirable event evolution relationships according to the *static thresholding* model described below.

$$\mathbf{G} = (\mathbf{V}, \mathbf{L}) \qquad (6)$$

where,

$$L' = \left\{(\varepsilon_i, \varepsilon_j) \mid conf\left((\varepsilon_i, \varepsilon_j)\right) \geq \lambda\right\} \quad (\varepsilon_i,\, \varepsilon_j \in V \text{ and } i \neq j) \qquad (7)$$

4. EXPERIMENTS

We have tested our proposed model on a corpus of news stories extracted from the CNN Newswire. All stories are written in English. The corpus is generated by automatic crawling and searching with the support of filtering by human annotator. There are totally 10 topics and 782 news stories in the corpus. The average length of documents is 582 words.

To avoid the errors generated by automatic clustering techniques and hence better compare the performances, we directly use the events manually clustered by human annotators as the test set. For the evaluation measures, we stick to the traditional measures of precision and recall and interpolate these rates to the standard 11 levels in the precision and recall graph.

Figure 2 shows the precision and recall curves of our experimental results. Our proposed model is the first approach (*EventEvolutioniGraph+StaticThresholding*) and the baseline is the third one (*EventThreading+BestSimilarity*) as claimed to perform best in Nallapati, et al. [2].

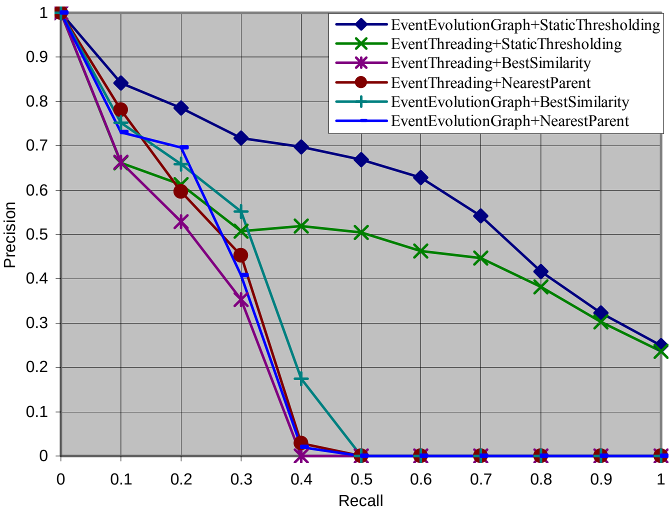

Figure 2. The Precision and Recall Curves (Interpolated to Standard 11 Levels) of the Comparative Experimental Results

5. CONCLUSIONS

In this paper, we have proposed the event evolution graph to model the evolution structure of news events inside a news affair in order to facilitate users' information browsing tasks. We utilized the cosine similarity of event term vectors as well as two decaying factors, i.e. temporal proximity and document distributional proximity, to measure the relatedness of news events. Finally, we showed in the experimental evaluation that our model outperforms rival models and the baseline model substantially.

6. REFERENCES

[1]. Juha Makkonen. Investigations on event evolution in TDT. In *Proceedings of HLT-NAACL 2003 Student Workshop*, pages 43–48, 2004.

[2]. Ramesh Nallapati, Ao Feng, Fuchun Peng, James Allan. Event threading within news topics. In *Proceedings of the 2004 Thirteenth ACM conference on Information and knowledge management (CIKM)*. pages 446-453, 2004.