ISSN:0975-9646

Episode Discovery from Event Evolution

Rutuja Ahirrao
Dept. of Information Technology
PCST, Indore (M.P.)

Sachin Patel
Head of Dept. of Information Technology
PCST, Indore (M.P.)

Rakesh Pandit
Assistant Professor,Dept. of Information Technology
PCST, Indore (M.P.)

Abstract-Event evolution is the most challenging task now days. There are series of events which are incident in a sequence and event evolution is used to arrange and show these events in the exact way as they occur. Lack of appropriate integration between events information causes severe problem as they misguide users.For example due to some reason if any flight is delayed and after a span of time it is rescheduled and not updated by various resources, and now if user searches the information of that flight via internet he will not able to find the correct status. So, the major concepts based on event evolution are: Collection of events, Validation of events instantly and Publishing of events. If events collected from various sources are not validated properly the incorrect information willbe reached to the user and the verified events must be published instantly so that user can access most recent document. This paper proposes a model forevent evolution and generates a graph for events as they occur.

Keywords-TDT, Event Evolution Graph, Event Tracking.

1. Introduction

Now a day's almost all news channels published their news in electronic versions also. Users of online news are increasing rapidly due to lake of time for reading hard copy of newspapers. Theses news also is available on various search engines and they will update the recent events as soon as they occur. Updating news straight away is the major issue and lots of research is continuously performed in this field. However, it also generates tremendous volume of news text stream. Managing interpreting, and analyzing such a huge volume of information is a difficult task. Techniques that are capable of extracting the underlying structure of the news events are desired. They are helpful for user to understand the evolution of events on the same topic. Event is something that happens at some specific time. Although user are able to capture the major events. There are several techniques present for event evaluation, this paper focused on TDT (Topic Detection and tracking).
TDT are used to discover topic wise data and the information is collected from various sources and may be from different language The TDT technique have been attempting to detecting or clustering news stories into these events, without defining or interpreting the association between these event. To present the development process of incident, we must model this kind of associations between events, which we define as event evolutions. This paper focused on all major components of event evaluation and shows the future direction in this field.
Section II focused on work done in the field of event evolution, Section III based of various challenges in the field of Event evolution and proposed model for efficient event evolution, Section IV is a conclusion part.

2. Background

Due to the popularity of the Internet, most news stories have electronic versions published on newswires. Retrieving news of the same topic from multiple sources and keeping information updated becomes more convenient and easier. Techniques that are capable of extracting the underlying structure of the news events are desired. They are helpful to understand the evolution of events on the same topic. The most important task in our proposed system is to construct the event evolution graph for identifying the event evolution relationships from the events.TDT is used to detect topics and tracking all necessary information related to it.

2.1 TDT:

In TDT, collected documents which contain information related to events are stored in hierarchical form. The main advantage of data store in hierarchical form is it is very easy to search data and time complexity of hierarchical data is less than linear data.Data in TDT is stored in the form of tree. There must be one root node which shows/indicate specific topic for news.
For example, movies, bollywood, jokes, etc.The subparts of root are various events which belong to that category.
Fig1 shows the example for TDT , the root for TDT is news, news are divided into various topics, Example Jokes,bollywood,movies and recipe then the subtopics are also divided into another level known as events. The leaf node of TDT tree is episode which contains main data.TDT is used to show events only not the sequence of events or relationship between them.

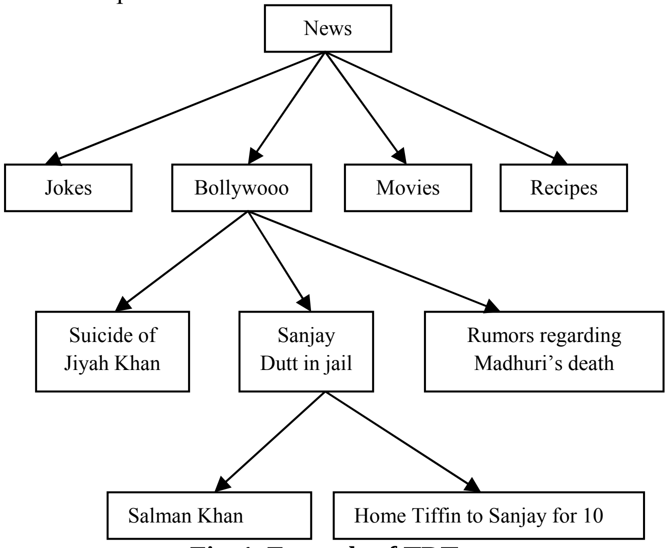


Fig. 1. Example of TDT

2.2 Event Evolution Graph:

Event Evolution Graph represents the relationship between events. Events are some process which are some process which happens at some specific time.TDT technique uses to detecting or clustering new stories into these events, but not focused on the association between the events. The relationships between events are important in event evolution because it shows the sequence of events. Event evolution graph is used to show relationship between events in well defined structure figure 2 shows event evolution graph.

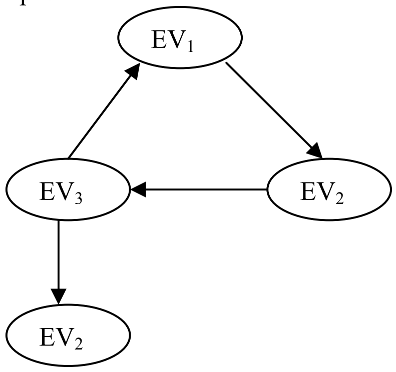


Fig. 2. Event Evolution Graph

Application of Event evolution graphs:

1. We may integrate the event evolution graphs with automatic summarization and named entity recognition techniques to provide a well-equipped web news information agency.
2. The named entity recognition techniques extract the names of persons or organizations and locations involved in an event. As a result, users can easily track the persons, organizations.
3. The interconnectivity of the graph structure can also support the construction of a convenient information-browsing platform for users.

Event episode Identification:

Event Episode Identification is a technique to identify events and episodes related to them. There are several steps present for this task:

a) Feature Extraction.
b) Feature Selection.
c) Document Representation
d) Document Clustering.

1. Feature Extraction:

In feature extraction technique system parses the documents in each document sequence to produce a list of nouns and noun phrases that exclude a set of pre-specified stop words.

2. Feature Selection:

For each document sequence, the feature selection phase select the top k-feature with the highest feature selection metric score to represent the documents in the document sequence.

3. Document Representation:

In the document representation phase, each document is then representing using the representative feature selected for the document sequence to which that document belongs.

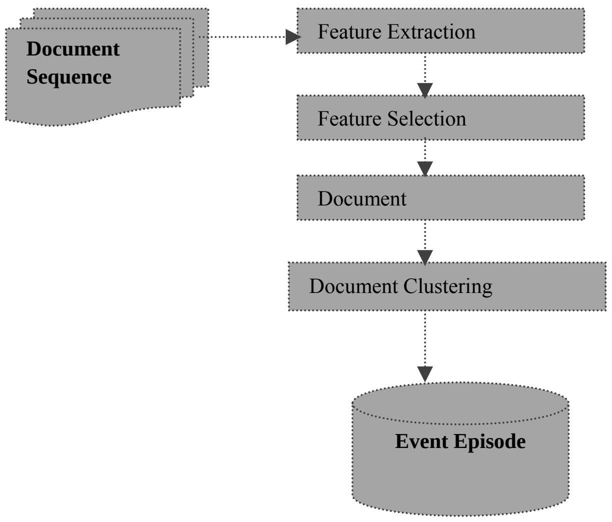


Fig. 3. Example of TDT

4. Document Clustering:

The documents clustering phase generates episode as cluster of documents. A clustering algorithm is based on a measure of the similarity between the documents to be clustered and similarity is based on the representing feature vector.

2.3 Event Relationship Construction:

The Input for event relationship construction will be the Output of the first phase i.e. Event Episode Identification. This phase mainly include three sub-phases:

A. Modeling Event evolution Relationship
B. Pruning Event Evolution Graph
C. Evolution Measures.

It should be noted that there are some online sources of well – generated news events. These sources after pre-processing, can serve as input to our proposed event evolution technique.

A. Modeling Event Evolution Relationships:

In modeling event evolution relationships, we propose to utilize

1 Vectors space
2 Event term vector
3 Temporal Proximity,
4 Document distributions proximity.

1. Vector space:

The vector space model use to measure the relatedness of events.

2. Event term vector:

In this case, we treat an event as a collection of stories/event episodes and view the information content of that event as the average of the term vectors of its stories. Thus, the content similarity between two events is the cosine similarity of their event term vectors.

3. Temporal proximity:

If two events are temporally close, an event evolution between them is more likely to exist. We use the temporal proximity between events to measure their relative temporaldistance between two events.

4. Document Distributional Proximity:

The document distributional proximity is similar to temporal proximity except that it substitutes the time with the distribution of documents.

B. Pruning Event Evolution Graphs:
Given a complete graph with directed edges for each pair of events, a pruning method removes those directed edges corresponding to invalid or weak event evolution relationships and generates an event evolution graph.
There are three pruning methods, namely,
1. Static Thresholding.
2. Static Pruning.
3. Dynamic pruning.

The pruning of directed edges depends on the value of event evolution scores and the degree of event threading and event joining.

C. Evaluation Measures:
We adopt the measurement of precision and recall for our evaluation.
1) **Precision (P)**: It is the ratio of the number of true and valid event evolution relationships retrieved by the automatic system to the total number of event evolution Relationships retrieved by the automatic system.
2) **Recall(R)**: It is the ratio of the number of true and valid event evolution relationships retrieved by the automatic system to the total number of true and valid event evolution relationships interprets manually.

2.4 DATASET
The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. To the best of my knowledge, it was originally collected by Ken Lang, probably for his Newsweeder: Learning to filternetnews paper, though he does not explicitly mention this collection. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering.

2.5 ORGANIZATION OF DATA
The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g.comp.sys.ibm.pc.hardware/comp.sys.mac.hardwar), while others are highly unrelated (e.g misc.forsale / soc.religion.christian). Here is a list of the 20 newsgroups, partitioned (more or less) according tosubject matter.

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Fig. 4. Organization of Data

2.6 EVOLUTION PATTERN MINING
EP mining phase aims to discover maximal temporal graph among all frequent temporal graph. Each generalized event episode obtained from the intersequence episode generalization from intersequence episode generalization is first assigned a distinct label, which we use to substitute for those documents that pertain to the event episode in each document sequence. Then we will construct a temporal graph for each document sequence and adopt the temporal pattern discovery algorithm to discover EPs from the set of temporal graph. The temporal graph represent temporal relationship in organize event episode in sequence according to their temporal order, but the temporal relationship do not necessarily reflect evolutions between event episodes. In evolution relationships, the temporal order is considered, but here we consider event content similarity, temporal proximity, and document distributional proximity.
In modeling event evolution relationships, we propose to utilize
1. Vectors space
2. Event term vector
3. Temporal Proximity,
4. Document distributions proximity.

2.6.1. Vector space:
The vector space model use to measure the relatedness of events. A k-term vector for S is denoted as $\omega = \omega_1, \omega_2 \ldots, \omega_k$. Let a story ibe represented as a weight end term vector $\omega_i = \omega_{i1}, \omega_{i2}, \ldots, \omega_{ik}$. On the basis of the traditional TF-IDF function, $\omega_{ik}$is defined as

$$\boxed{\omega ik = \frac{\mathrm{tf}_{ik}}{\max \mathrm{tf}_{il}} \log \frac{N}{dfk}}$$

wheretf$_{ik}$= It is the frequency of term k in the news document i,
N = It is the total number of news documents in that topic,
df$_k$= It is the number of news documents in that topic contain in termk,and
maxltf$_{il}$ = It is the maximum term frequency for all terms in document i.
If we only consider the TF factor only, ω_{ik}becomes

$$\boxed{\omega ik = \frac{\mathrm{tf}_{ik}}{\max \mathrm{tf}_{il}}}$$

2.6.2 Event term vector:
The event term vector of event $e_j$using the average of the term vectors of stories that belong to $e_j$as: $\omega'_j = <\omega'_{j1}, \omega'_{j2,\ldots\ldots}\omega'_{jk}>$, where

$$\boxed{\omega'_{jk} = \frac{1}{n_j} \sum_{\forall s_i \in e_i} \omega_{ik}}$$

where$n_j$is the number of stories in S that belongs to event e_j.

The event content similarity is then defined as follows:

$$\boxed{\cos\_\mathrm{sim}(e_i , e_j) = \frac{\sum_{x=1}^{k} \omega'_{ix}\, \omega'_{jx}}{\sqrt{\sum_{x=1}^{k} (\omega'_{ix})^2 + \sum_{x=1}^{k} (\omega'_{jx})^2}}}$$

In this case, we treat an event as a collection of stories/event episodes and view the information content of that event as the average of the term vectors of its stories. Thus, the content similarity between two events is the cosine similarity of their event term vectors.

2.6.3Temporal proximity:

If two events are temporally close, an event evolution between them is more likely to exist. We use the temporal proximity between events to measure their relative temporaldistance between two events, defined as the following decayingfunction:

$$\boxed{tp(e_1,\ e_2) = e^{-\alpha[\frac{d(\tau(e_1),\tau(e_2))}{T}]}}$$

whereT is the event horizon defined as the temporal distance between the start time of the earliest event timestamp and the end time of the latest event timestamp in the same topic. αis the time decaying factor which is between zero and one.

2.6.4Document Distributional Proximity:

$$\boxed{df(e_1,\ e_2) = e^{-\beta\frac{m}{n}}}$$

Where m = It is the number of documents that belong to the events happening in-between event e_1 and e_2.

N = It is the total number of documents in the topic.

The document distributional proximity is similar to temporal proximity except that it substitutes the time with the distribution of documents. The event content similarity, temporal proximity, and document distributional proximity, we formally propose our event evolution scoring function to estimate the likelihood that an event evolution relationship exists from an event e_1 to another event e_2 as

score((e_1,e_2))=0 if $\tau(e_i)$ $\tau(e_j)$;

Cos-sim(e_1,e_2)*tp(e_1,e_2)*df(e_1,e_2) if $\tau(e_i) \rightarrow \tau(e_j)$.

The EP mining phase aims to discovered maximal temporal graph among all frequent temporal graphs. Each generalized event episode obtained from intersequence episode generalized phase is assigned a distinct label, which we use to substitute for those documents that pertain to the event episode in each document sequence. Subsequently we construct a temporal graph in each documents sequence and adopt the temporal pattern discovery algorithm [8] to discover EPs from the set of temporal graph. The temporal pattern discovered downward closure property of the support measure, which improve the efficiency of searching for frequent temporal graphs and adopts an iterative procedure .For given a set of process instances, the temporal pattern discovery is to findthe maximal temporal graphs among all frequent temporal graphs. Each such temporal graph isreferred to as a temporal pattern.

2.7 TEMPORAL PATTERN DISCOVERY ALGORITHMS:

There are three different algorithms, namely

1. TP-Graph,
2. TP-Item set,
3. TP-Sequence,

are proposed for the described temporal pattern discovery problem.

TP-Graph precedes its discovery process directly based on the temporal graph representation.

2.7.1 TP-Graph Algorithm:

As with association rule (Agrawal and Srikant, 1994) and sequential pattern (Agrawal and Srikant, 1995) algorithms, the TP-Graph algorithm exploits the downward closure property of the support measure to improve the efficiency of searching for frequent temporal graphs. In algorithms, potentially frequent temporal graphs (or called candidate temporal graph) of size k are constructed by joining frequent temporal graphs of size k−1. The process instances are then scanned to identify frequent temporal graphs of size k from the set of candidate temporal graphs of the same size. The resultant frequent temporal graphs are then used to prune the non-maximal frequent temporal graphs derived in the previous iteration (i.e., k−1). This
procedure is iteratively executed until no further frequent temporal graphs can be found.

Let Ck and Lk denote the set of candidate temporal graphs and the set of frequent temporal graphs ofsize k, respectively. Each iteration k performs the following three steps whose challenges andsolutions are detailed in the following subsections, respectively.

1. If k=1, Ck is the set of all single-activity temporal graphs. Otherwise, Ck is generated by joining in pair-wise the frequent temporal graphs of size k−1(i.e., Lk - 1).
2. Scan the process instances to determine Lk from Ck.
3. If k>1, prune from Lk-1 all non-maximal temporal graphs that are temporal subgraphs of any temporal graph in L k.

The previously described downward closure property can further be exploited to reduce the set
of resulting candidate temporal graphs. A candidate temporal graph G of size k will not be
frequent if any of its temporal subgraphs of size k−1 is not in Lk-1 and, hence, should be
eliminated from Ck. Such pruning process requires, for each candidate temporal graph of size k,
the derivation (using the subtraction operation defined in Definition 10) of all of its temporal
subgraphs of size k–1. The pseudo code of GenerateCandidateGraph() for generating a set of
candidate temporal graphs of size k from a set of frequent temporal graphs of size k−1 and that
ofDeriveSubgraph() for deriving all temporal subgraphs of size |G|−1 for a temporal graph G
are listed below:

```
DeriveSubgraph(a temporal graph: G): a set of temporal graphs
{
        Subgraph = Ø;
        For (each vertex v in G) {
        Source = the set of vertices incident to v;
        Sink = the set of vertices incident from v;
        SG = G – {v};
        For (each vertex pair (vs, vd) where vs ∈ Source and vd ∈ Sink) {
        If there does not exist a path between vs and vd in SG then SG = SG ∪{vs→vd};
        } /* end-for */
        Subgraph = Subgraph ∪ {SG};
        }
        Return Subgraph;
        }
```

```
Generate CandidateGraph (a set of frequent temporal graphs: TGS): a set of temporal graphs)
{
        CandidateSet = Ø;
        For (each pair of graphs (Gi, Gj) in TGS) {
        For (each source vertex s in Gi) {
        For (each sink vertex e in Gj) {
        If (s ≠ e and Gi−{s}= Gj−{e}) { /* joinable */
        UG1 = Gi ∪Gj; UG2 = Gi ∪Gj ∪ {s→e};
        CandidateSet = CandidateSet ∪ {UG1};
        If there exists no path from s to e in UG1
        Then CandidateSet = CandidateSet ∪ {UG2};
        } /* end-if */
        } /* end-for */
        } /* end-for */
        } /* end-for */
        For (each graph G in CandidateSet) {
        If DeriveSubgraph(G) ∩ TGS ≠ DeriveSubgraph(G)
        Then CandidateSet = CandidateSet – {G};
        } /* end-for */
```

2.7.2 Pruning Non-maximal Temporal Graphs:

Apparently, if a temporal graph G is frequent, all of its temporal subgraphs also are frequent.

According to Definition 9, the temporal subgraphs of G are not maximal and should be pruned since they reveal less information on frequent activities and temporal relationships than G. To retain only maximal temporal graphs, PruneSubgraph() is invoked at each iteration to eliminate any non-maximal temporal graphs obtained at the previous iteration.

/* TGSn is the set of frequent temporal subgraphs obtained at iteration n */

```
PruneSubgraph(a set of graphs: TGSn-1, a set of graphs: TGSn): a set of temporal graphs
{
        For (each graph G in TGSn) {
        SG = DeriveSubgraph(G)4;
        TGSn-1 = TGSn-1 − SG;
} /* end-for */
        Return TGSn-1;
```

Upon completion of the EP mining phase, the EP technique generated a set of EPs and feature vectors of event episodes that are generalized from a collection of document sequences.

3. Event tracking

We extend a traditional ET technique and propose the EP-supported ET (EPET) technique that employs the EPs discovered by the EP technique. As a feature co-occurrence approach, the traditional ET technique encounters a dilemma that can diminish its tracking effectiveness. To address the dilemma faced by the ET technique, we believe that discovering common event EPs whose application can facilitate ET is appealing and essential.

The EPET technique consists of two processes:

1. Discovery of New events
2. Tracking of events.

1. Discovery of New Events:

Given a set of historical document sequences of the same event type as the target event, the discovery process is as shown in fig.[7] finds EPs and the feature vectors of event episode using proposed EP technique. Subsequently, the event episode that appears in each sample news story will be identified in the episode identification phase. A sample news story first is represented with the k most representative features according to the TF-IDF scheme. The similarity between a news story and an event episode obtained by the EP technique is determined by the co-sine similarity measure.

For each sample news story , if its similarity to any event episode is grater than or equal to s user-defined threshold for episode identification ,the event episode for which the largest similarity is attained is considered as the episode discussed in this sample news story; otherwise no event episode is assigned. Consequently, the progress of the target event, which is described by the sequence of event episodes of the sample news stories is then stored for use by tracking process of the EPET technique.

2. Tracking of events:

The tracking process of EPET technique fig.(5) determines whether an incoming news story pertains to the same event as sample news stories. As with the ET technique, the feature vector of new news story is compared with query for the sample news stories. If its similarity is less than a prespecified. Tracking similarity threshold, the new news story is assumed to pertain to a different event, and the tracking process terminates. Otherwise , filtering that is based on the previously discovered. EPs is activated. In this case ,the event episode in the new news story is identified first using the episode in the new news story is identified first using the episode identification method in the discovery process of the EPEY technique. Assuming that the new news story discussed the same event as that of the sample news stories ,the progress of the event after the addition of the event episode of the new news story must violate any EP discovered in the discovery process

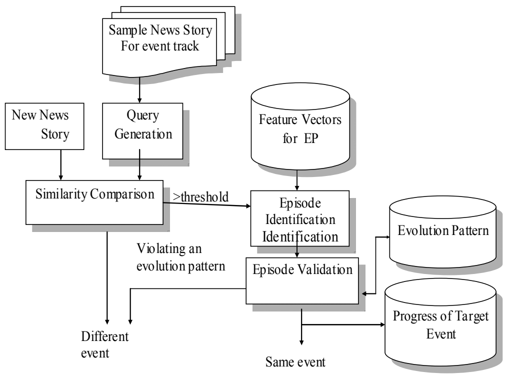


Fig. 5. Tracking Process of the EPET technique

4. Challenges and Future Directions

Event Evolution can be performed using TDT & Event evolution graph but the major part of event evolution are events. If information is not collected properly for each event then we can't find out the efficient output for Event Evolution. The major challenges of Event Evolution are: Collection of events, Validation of events and Publishing of an event.

Collection of Events:

The central concept of event evolution is events and the result efficiency based on event collection. For any incident the whole information (event) must be collected because each and every event has same priority.

Validation of Events:

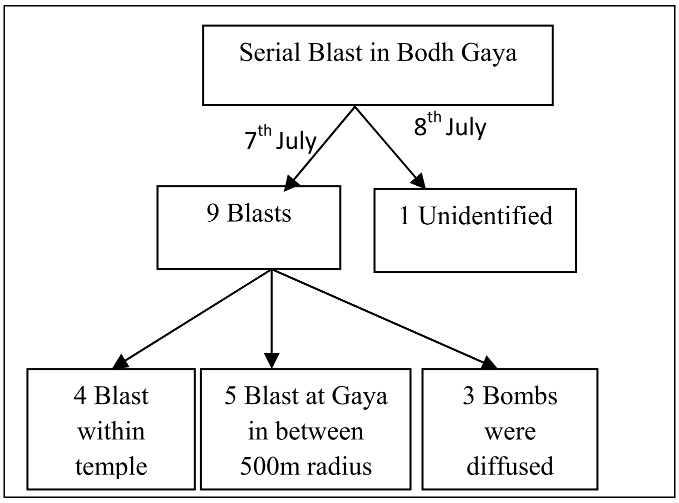
All collected events must be validated properly because any incorrect information may change the scenario of event.

Publishing of event:

When events are supposed to be published they must be published properly and in a proper sequence

Example1 shows the event evolution process for Bodhgaya bomb blast. This example covers all the steps (phases) which are involved in Event Evolution.

Example: Serial Blast in Bodh Gaya



Serial Blast in Bodh Gaya

Mahabodhi Temple Blast around the Mahabodhi temple complex and UNESCO world heritage site in Bodh Gaya: (7th July 2013)

On 7th July 2013, ten bombs exploded between 5:30 and 6.00 IST, at that time in the temple SulaChantry and meditation was running, because this time was the beginning time of their daily routine. Mahabodhi Mahavihara is an ancient and most famous temple for Buddhist. In this serrate place there is an holy Bodhi tree where Gautama Buddha is believed to have attained enlightenment.

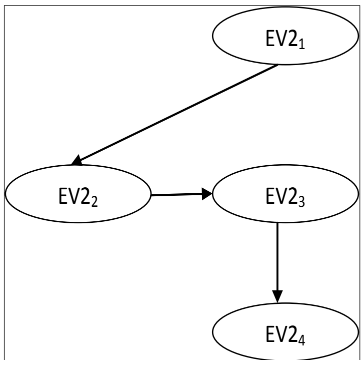
There are 10 bomb blasts that occurred and 3 bombs got diffused by bomb disposal squard.The main incidents of these blasts have occurred in following sequence:

EV1: On 7th July 2013, the news regarding bomb blast in Mahabodhi Temple complex spread out.
EV2: News regarding 9 bomb explosions has founded.
EV3: There were 4 blasts occurred within the Mahabodhi temple complex.
EV4: There were 5 bomb blasts occurred in Gaya,in a 500-meter radius of Mahabodhi temple complex.
EV5: 3 bombs were diffused by bomb disposal sqaurd.
EV6: On 8th July 2013, Indian home minister Mr.Sushil Kumar Shinde said that theer had actually been ten blasts.

Sequence of each events i.e. EV_1 to EV_6 can be divided multiple events or subevents .

EV_3 can be divided into following events:

// **Event 3**: News regarding 9 bomb explosion has founded.
EV3$_1$: The first bomb exploded at 5:30 IST.
EV3$_2$: 2 minutes later, next bomb exploded.
EV3$_3$: Third bomb exploded on southern side
EV3$_4$: Fourth bomb exploded in the northern side of the complex.



//**Event 4**: 5 blasts occurred within the Mahabodhi temple complex in a 500-meter radius.

EV4$_1$: A low intensity bomb exploded
EV4$_2$: Three bombs exploded at tergar (Tibetan) monastery.
EV4$_3$: One bomb exploded on a bus parked at Sujata Bypass.

//**Event 5:** three bombs were diffused by bomb disposal squads.

$EV5_1$: Three bombs were diffused by bomb disposal squads.

//**Event 6:** actual 10 blasts.

On 8th July 2013, Indian home minister Sushilkumar Shinde said that there had actually been 10 blasts.

Due to the popularity of the Internet, most news stories have electronic versions published on newswires. Retrieving news of the same topic from multiple sources and keeping information updated becomes more convenient and easier. Techniques that are capable of extracting the underlying structure of the news events are desired. They are helpful to understand the evolution of events on the same topic. The most important task in our proposed system is to construct the event evolution graph for identifying the event evolution relationships from the events.

In our proposed system first we identify event episodes for given a set of document sequence D pertaining to the same event type. Certain event episodes within D occur frequently and possess specific temporal relationships. After identification of an event episode we construct an event evolution graph. An event evolution graph is a directed acyclic graph (DAG) consisting of events as the nodes and event evolution relationships as the directed edges between nodes. Given a set of n distinct news stories(here we story is referred as an event episode) $S = \{s_1, s_2,\ldots,s_n\}$ on a given news topic, we have a set of m events $E = \{e_1,e_2,\ldots.,e_n\}$ and their event timestamps $T = \{t_1,t_2,\ldots.t_n\}= \tau(e_i)$. Each story is assigned to one of the m events. A directed edge from vertex $e_i$to $e_j$is created in the event evolution graph if there is an event evolution relationship from $e_i$to e_j. Event e_i is the parent of event $e_j$and event e_i is the child of event e_i. L is the set of event evolution relationships, $L = \{(e_i, e_j)$ where $e_i, e_j \in E$. Therefore, the event evolution graph G is a directed acyclic graph, $G = \{E, L\}$.

5. Conclusion

Our proposed system discovering event episode together with temporal relationship that occurs frequently from sequence of documents. These uncovered EPs can be applied to several interesting application domains, including knowledge management and be used to fascinating existing document management and retrieval techniques. Proposed method contributes to ET research by addressing the dilemma – association with the event resemblance and feature shifting phenomenon faced by traditional ET techniques.

There is a large volume of news stories reporting ongoing incidents on the Web. In order to capture the development of the events in these incidents efficiently and effectively, we propose the event evolution identification technique to automatically identify event evolution relationships and represent the underlying structure as an event evolution graph. Contrary to the traditional view of topics as flat hierarchies in document clustering and categorizing tasks or tree structure in event threading , we view news topic as a DAG with events as its nodes and event evolution relationships as its directed edges. We will utilize the temporal proximity and document distributional proximity as decaying functions in addition to the cosine similarity of event term vectors for measuring event content similarities. The event evolution graphs are useful to present the underlying structure of the events extracted for a topic. It helps to understand how the events evolve along the timeline.

References:

[1] Rutuja Ahirrao and Sachin Patel. Article: An Overview on Event Evolution Technique. *International Journal of Computer Applications* 77(10):7-11, September 2013. Published by Foundation of Computer Science, New York, USA.

[2] J. Allan, R. Papka, and V. Lavrenko, “On-line new event detection and tracking,” in Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Development Inf. Retrieval, Melbourne, Australia, 1998, pp. 37–45.

[3] Y. Yang, T. Pierce, and J. Carbonell, “A study on retrospective and online event detection,” in Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Development Inf. Retrieval, Melbourne, Australia, 1998, pp. 28–36.

[4] J. Carthy, “Lexical chains for topic detection,” Dept. Comput.Sci.,Univ.College Dublin—National Univ. Ireland, Dublin, Ireland, 2002. Tech. Rep.

[5] J. Makkonen, “Investigations on event evolution in TDT,” in Proc. Conf. North Amer. Chapter Assoc. Linguistics Human Language Technol., HLT-NAACL Student Research Workshop, Edmonton, AB, Canada, 2003, pp. 43–48.

[6] R. Nallapati, A. Feng, F. Peng, and J. Allan, “Event threading within news topics,” in Proc. 13th ACM Int. Conf. Inf. Knowl. Management, Washington, DC, 2004, pp. 446–453.

[7] Wei and Y. Chang, “Discovering event evolution Graph from corpora,” IEEE Trans. Syst., Man, Cybern.A, Syst., Humans, vol. 39, no. 4, July. 2009.

[8] Wei and Y. Chang, “Discovering event evolution patterns from document sequences,” IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 37, no. 2, pp. 273–283, Mar. 200