EventRiver: An Event-Based Visual Analytics Approach to Exploring Large Text Collections with a Temporal Focus

Dongning Luo, Jing Yang, Milos Krstajic, William Ribarsky, and Daniel Keim

Abstract—Many Text Collections with a Temporal Focus (TCTFs), such as news corpora and weblogs, are generated to report and discuss real life events. Thus Event-Related Tasks (ERTs), such as detecting the real life events driving the text, tracking their evolution, and investigating the reports and discussions around these events, are important when exploring such text collections. In this paper, we propose a novel visual analytics approach named *EventRiver* to help users conduct ERTs that incorporates and leverages human efforts. EventRiver employs a novel incremental streaming text clustering algorithm to detect clusters with temporal locality from TCTFs, which can be mapped to real life events driving the text generation. The clustering is based on a streaming data model so that newly arrived documents can be processed without re-processing the existing documents. The semantics of the clusters, their strengths over time, and the relationships among the clusters are automatically captured and visually presented to users. The users can then conduct the ERTs using the rich set of interactions provided in EventRiver. A working prototype of EventRiver has been implemented for exploring news corpora. A set of user studies, experiments, and a formal user study have been conducted to evaluate its effectiveness and efficiency.

Index Terms—Visual Analytics, Information Visualization, Topic, Event, Text, Clustering.

1 Introduction

As Jon Kleinberg stated, the *alignment* problem is an issue in many contexts of on-line information stream exploration, which is described as to *align the virtual events found in an information stream with a parallel set of events taking place outside the stream, in the real world* [1]. We argue that this alignment problem is extremely important when exploring *Text Collections with a Temporal Focus (TCTFs)*, namely text collections into which new contents are continuously added over time and have meaningful time stamps that are critical when analyzing the collections, such as news corpora and weblogs. The reason is that the purposes of many TCTFs are to report and discuss real life events (an *event* refers to an occurrence that happens at a specific time and draws continuous attention [2]) in a timely manner. To understand such TCTFs, it is necessary *to detect the real life events motivating the text generation, to learn their semantics and temporal context, and to track their evolution over time* [1]. It is also important for users to *find documents about events of interest and investigate them in full details* [1]. We refer to all the above tasks as *Event-Related Tasks (ERTs)*. Since such TCTFs are important information sources in a wide variety of applications, such as social and cultural studies, government intelligence, and business decision making, and it is effort-extensive to explore them manually, there is an urgent need for tools that aid human beings in conducting the ERTs on TCTFs effectively and efficiently.

Towards this goal, many efforts have been made to automatically detect and track events in TCTFs under the name of Topic Detection and Tracking (TDT) [3], [4]. Unfortunately, few efforts have been made to intuitively and effectively convey the automatic analysis results to users and to allow the users to interactively explore the results for in-depth analyses.

There also exist a range of visualization systems for exploring TCTFs. However, most of them do not support the ERTs directly. For example, ThemeRiver [5] allows users to follow the strengths of significant keywords over time, but individual keywords are not capable of explicitly revealing events and their semantics and evolution. IN-SPIRE [6] reveals topics in a large text collections using clustering. However, since time stamps of the documents are not considered in clustering, IN-SPIRE displays permanent topics rather than bursty events. Its temporal animation controlled by a time slicer helps reveal bursty events, but it is hard to learn the temporal context of the events and follow their evolution due to change blindness [7].

In this paper, we propose a novel visual analytics approach to addressing the above challenges. It aims to visually support the following ERTs on TCTFs driven by real life events:

- **Event Browsing:** To allow users to detect the major events and the big stories consisting of multiple related events that motivate the TCTFs without prior knowledge, and to learn their semantic information,

- D. Luo, J. Yang, and W. Ribarsky are with the University of North Carolina at Charlotte, USA;
- M. Krstajic and D. Keim are with the University of Konstanz, Germany.

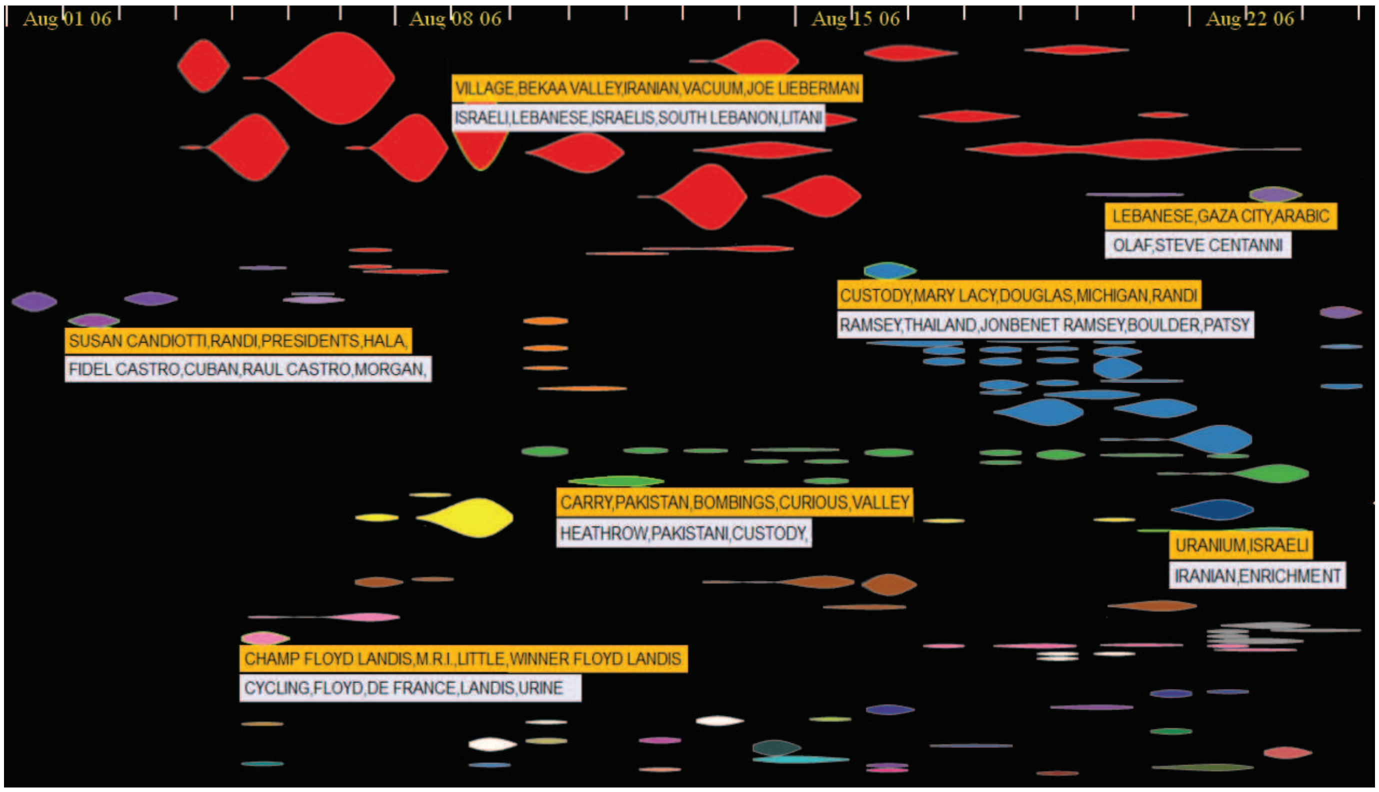

Aug 01 06
Aug 08 06
Aug 15 06
Aug 22 06
VILLAGE,BEKAA VALLEY,IRANIAN,VACUUM,JOE LIEBERMAN
ISRAELI,LEBANESE,ISRAELIS,SOUTH LEBANON,LITANI
LEBANESE,GAZA CITY,ARABIC
OLAF,STEVE CENTANNI
CUSTODY,MARY LACY,DOUGLAS,MICHIGAN,RANDI
RAMSEY,THAILAND,JONBENET RAMSEY,BOULDER,PATSY
SUSAN CANDIOTTI,RANDI,PRESIDENTS,HALA,
FIDEL CASTRO,CUBAN,RAUL CASTRO,MORGAN,
CARRY,PAKISTAN,BOMBINGS,CURIOUS,VALLEY
HEATHROW,PAKISTANI,CUSTODY,
URANIUM,ISRAELI
IRANIAN,ENRICHMENT
CHAMP FLOYD LANDIS,M.R.I.,LITTLE,WINNER FLOYD LANDIS
CYCLING,FLOYD,DE FRANCE,LANDIS,URINE

Fig. 1. *CNN* news from Aug. 1 to 24, 2006 ($29,211$ closed-caption documents) in EventRiver. TLCs with small strengths were filtered out.

temporal context, and the attention received without reading the documents.

- **Event Search, Tracking, and Association:** To allow users to search events of interest by keywords or example text, to track the evolution of an event of interest, and to examine the possible relationships among multiple events within the temporal context.
- **Event Investigation:** To allow users to examine the documents about an event of interest in full detail.

Our approach tightly integrates automated text analyses with visualizations and interactions. It automatically extracts clusters that can be mapped to real life events from TCTFs using text analysis techniques. It then visually presents the event-related clusters to users so that they can interactively conduct the ERTs on the display. Therefore, the world view gap, namely *the gap between what is being shown and what actually needs to be shown to draw a straightforward representational conclusion for decision making* [8], is filled. Since the event-related clusters are displayed in a river-like view in which the temporal information of the events and the continuous attention they draw is explicitly depicted (see Fig. 1 for a screenshot), our approach is named *EventRiver*.

EventRiver integrates several novel automatic analyses. First, an incremental streaming text clustering algorithm is applied. It automatically detects *Temporal Locality Clusters (TLCs)*, namely clusters containing documents that are *coherent in content and adjacent in time*, from TCTFs. Each TLC maps to a real life event and the continuous attention it draws. The clustering algorithm processes new updates of a constantly evolving text collection without reprocessing the existing text. Built upon this algorithm, EventRiver has the potential to process large constantly evolving text collections in real time. The TLCs are further clustered into TLC groups by their content. A TLC group maps to a big story consisting of multiple related events. Users can track event evolutions and detect related events within TLC groups. Second, an automatic semantic information extraction approach is used to generate a semantic representation for each TLC. It extracts the unique content and the semantic context of a TLC to help users grasp the semantics of its motivating event effectively and efficiently.

EventRiver intuitively presents the automatic analysis results to users and allows them to interactively conduct the ERTs on the visualization. Fig. 1 shows a screenshot of the EventRiver visualization. It looks like a river of bubbles and each bubble represents a TLC. The horizontal axis of the river is a time axis where time flows from left to right. The width of a bubble along the vertical axis indicates the number of documents in the TLC that are released at the time. Most bubbles have pointy left ends and bulb-shaped bodies. The pointy left ends represent the earliest reports of the motivating events. The bodies represent the documents reflecting the continuous attention drawn to the events. The vertical positions and the colors of the bubbles represent the relationships among the TLCs. Bubbles with the same colors and adjacent to each other are TLCs in the same TLC group.

Users can browse the bubbles for *significant events and* *big stories driving the TCTF*. They can also *track event evolutions* by examining the TLC groups. The labels of the bubbles present the semantics of the TLCs. The users can click on a bubble to trigger a shoebox (see Fig. 2(b) for an example) in which the *documents related to the event can be interactively investigated*. They can also open a storyboard for a TLC group to *examine the event evolution in detail* (see Fig. 3 for an example). A rich set of interactions are provided to allow users to interactively navigate, search, track, associate, and examine the events.

A fully working prototype of EventRiver has been implemented. It has been used to explore large collections of close-captioned broadcast news videos. A set of case studies, experiments, and a formal user study have been conducted to evaluate the effectiveness and efficiency of EventRiver. The evaluation showed that EventRiver concisely presented significant events and their evolutions in large TCTFs without clutter, and allowed users to explore the rich information carried by the TCTFs in a manageable way.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 presents the automatic analysis techniques used in EventRiver. Section 4 introduces the visualizations and interactions for event browsing, retrieval, tracking, association, and detail investigation. Sections 5, 6 and 7 present the case studies, the experiments, and a formal user study we conducted to evaluate EventRiver. Section 8 concludes the paper.

2 Related Work

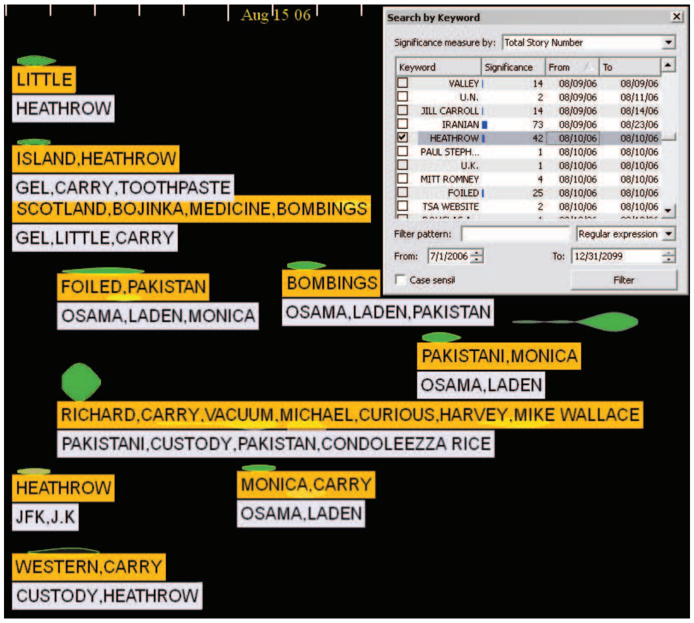
Many efforts have been made on Topic Detection and Tracking (TDT) [3], [4]. For example, Mei and Zhai [9] extract events using a language model-based approach and analyze the temporal and evolutionary structure of the events to discover the evolutionary theme patterns. Fung et al. [10] propose a *Time Driven Documents Partition* algorithm to construct an event hierarchy in a text corpus based on a given query. Allan [2] presents a survey of recent TDT work. TDT approaches usually focus on system-provided answers [11].

Most traditional visualization approaches for exploring TCTFs fall into one of two categories, namely *Keyword Tracing Techniques (KTTs)* and *Time Slicing Techniques (TSTs)*. KTTs visually depict the strength changes of individual keywords in a text collection over time to reveal its dynamic semantics. A representative approach in this category is ThemeRiver [5], which depicts the strength changes of individual keywords as currents within a river flowing along a time axis. LensRiver [12] is a variation of ThemeRiver. It reveals the global relationships among the keywords by constructing a keyword hierarchy for the whole text collection and bundling the currents according to the hierarchy. Narratives [13] uses time plots to visualize how concepts (keywords) change over time in weblog archives and introduces several methods to explore how these concepts relate to each other. It uses events to calculate the strengths of the keywords, where an event consists of the URL to a referring blog, the URL to a target article, and a set of relevant keywords, which has a different meaning from events mentioned in this paper.

KTTs do not visually present event information to users and support ERTs directly. The users have to speculate about events and their semantics, temporal features, and evolution by observing the strength changes of individual keywords. Since the same keyword can contribute to multiple events even at the same time period and the semantics of an event are usually conveyed by a set of co-occurring keywords in a short time period, this speculation is not only time consuming, but also inaccurate. In addition, to access documents related to an event, the users may need to conduct a set of searches.

Time Slicing Techniques (TSTs) divide a TCTF into multiple time slices, generate a static view for each slice, and dynamically display the static views in a time sequence to reveal the topic changes. For instance, Hetzler et al. [11] visualize TCTFs in a 2D projection space with fresh and stale documents visually distinguished. An evolving window of time is used to control the animation. Erten et al. [14] present a temporal graph layout algorithm to visualize the topic evolution of a computing literature collection. Textpool [15] buffers live text streams into a pool, extracts the most frequently occurring salient terms from the buffered streams, and displays them in a dynamic text collage. TagLine [16] characterizes the most interesting tags associated with a sliding interval of time and uses an animation to reveal how the interesting tags evolve over time. Although TSTs can be good at revealing topics in each time slice, users often suffer from change blindness [7] when investigating event evolutions.

There exist several visualization approaches that organize text into topics and display the topics along a time axis, including a couple of recent visual analytics approaches. *Conversation landscape* [17] groups postings in the same conversation together and displays them as horizontal lines along y axis representing time. CAST [18] applies a hierarchical agglomerative clustering algorithm on keywords extracted from a news corpus to generate themes (clusters of keywords). The themes are visualized as an array of the theme words over time with width-changing lines connecting words in the same theme to indicate the flow of stories. News Cycle [19] clusters textual variants of short, distinctive, and quoted phrases and uses a stacked plot to reveal the daily rhythms and their temporal patterns in news media and blogs. The analysis algorithms underlying the above approaches require reprocessing the whole text collections when new data arrive. The consistency among the results is not guaranteed after each reprocessing. Differently, EventRiver uses an incremental algorithm without requiring reprocessing the whole dataset when new data arrive.

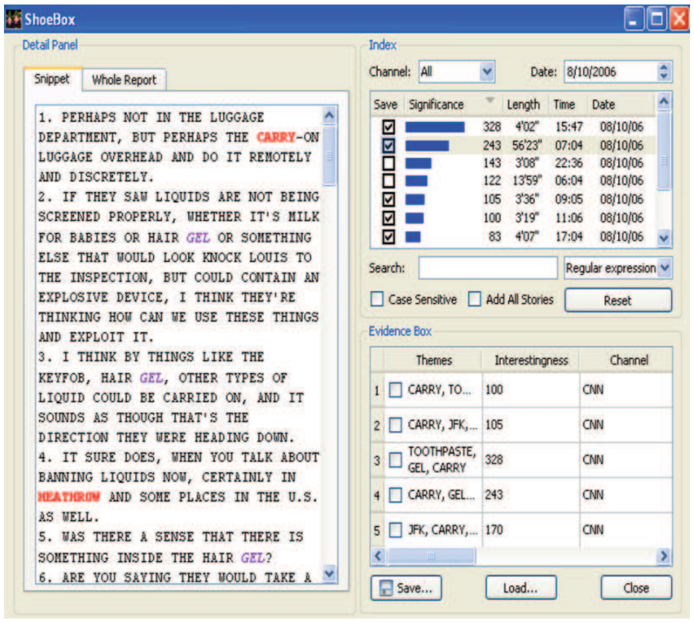


Significance measure by: Total Story Number

Keyword	Significance	From	To
☐ VALLEY	14	08/09/06	08/09/06
☐ U.N.	2	08/09/06	08/11/06
☐ JILL CARROLL	14	08/09/06	08/14/06
☐ IRANIAN	73	08/09/06	08/23/06
☑ HEATHROW	42	08/10/06	08/10/06
☐ PAUL STEPH...	1	08/10/06	08/10/06
☐ U.K.	1	08/10/06	08/10/06
☐ MITT ROMNEY	4	08/10/06	08/10/06
☐ FOILED	25	08/10/06	08/10/06
☐ TSA WEBSITE	2	08/10/06	08/10/06

Filter pattern: Regular expression
From: 7/1/2006 To: 12/31/2099

(a) Zooming into selected TLCs



1. PERHAPS NOT IN THE LUGGAGE DEPARTMENT, BUT PERHAPS THE CARRY-ON LUGGAGE OVERHEAD AND DO IT REMOTELY AND DISCRETELY.
2. IF THEY SAW LIQUIDS ARE NOT BEING SCREENED PROPERLY, WHETHER IT'S MILK FOR BABIES OR HAIR GEL OR SOMETHING ELSE THAT WOULD LOOK KNOCK LOUIS TO THE INSPECTION, BUT COULD CONTAIN AN EXPLOSIVE DEVICE, I THINK THEY'RE THINKING HOW CAN WE USE THESE THINGS AND EXPLOIT IT.
3. I THINK BY THINGS LIKE THE KEYFOB, HAIR GEL, OTHER TYPES OF LIQUID COULD BE CARRIED ON, AND IT SOUNDS AS THOUGH THAT'S THE DIRECTION THEY WERE HEADING DOWN.
4. IT SURE DOES, WHEN YOU TALK ABOUT BANNING LIQUIDS NOW, CERTAINLY IN HEATHROW AND SOME PLACES IN THE U.S. AS WELL.
5. WAS THERE A SENSE THAT THERE IS SOMETHING INSIDE THE HAIR GEL?
6. ARE YOU SAYING THEY WOULD TAKE A

Channel: All Date: 8/10/2006

Save	Significance	Length	Time	Date
☑	328	4'02"	15:47	08/10/06
☑	243	56'23"	07:04	08/10/06
☐	143	3'08"	22:36	08/10/06
☐	122	13'59"	06:04	08/10/06
☑	105	3'36"	09:05	08/10/06
☑	100	3'19"	11:06	08/10/06
☑	83	4'07"	17:04	08/10/06

Search: Regular expression

	Themes	Interestingness	Channel
1	☐ CARRY, TO...	100	CNN
2	☐ CARRY, JFK,...	105	CNN
3	☐ TOOTHPASTE, GEL, CARRY	328	CNN
4	☐ CARRY, GEL...	243	CNN
5	☐ JFK, CARRY,...	170	CNN

(b) A shoebox for the TLC labeled "GEL, LITTLE, CARRY"

Fig. 2. Investigating TLCs about the *2006 Transatlantic Aircraft Plot*. The TLCs in (a) were selected using the keyword *Heathrow*.

3 Automatic Analyses

3.1 TLCs and TLC Groups

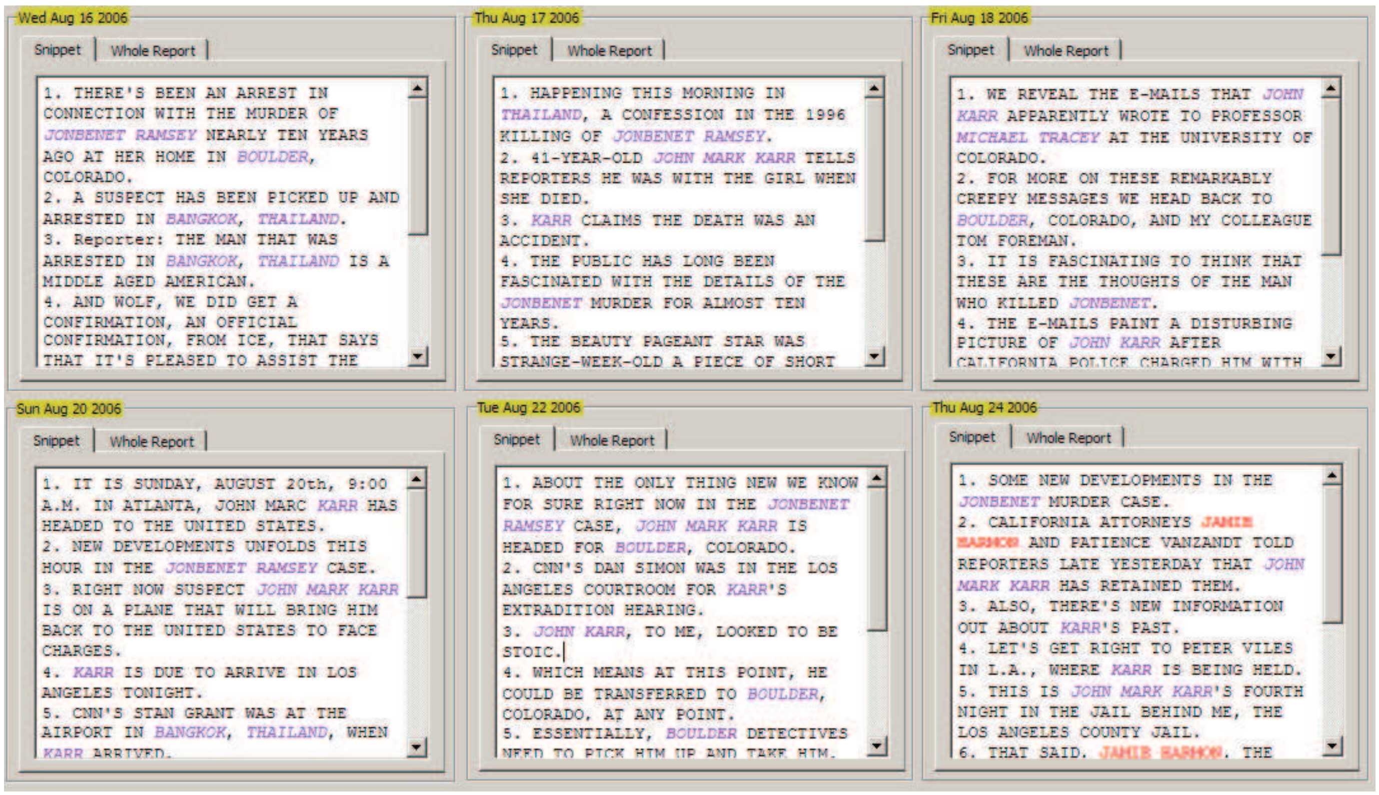
An event refers to *an occurrence that happens at a specific time and draws continuous attention* [2]. We organize documents motivated by the same real life event into a cluster and visually present the cluster to users so that they can be aware of the real life event. Our clustering is based on the following **document generation model**: once a real life event happens, documents about it will be generated during the time span when it draws continuous attention. They form a cluster of documents that *have closely related content* and *coincide or are adjacent in time*. Documents with similar content can also be generated a while after this time span, but they are usually motivated by new developments of this event, namely its follow-up events, and thus should be associated with the follow-up events rather than this event.

Since the time adjacency is important for the clusters we are looking for, we call them **Temporal Locality Clusters (TLCs)**. The mapping between a TLC and a real life event allows users to estimate what happens in the event, when the event occurs, how long it attracts continuous attention, and how significant the attention is according to the semantics, temporal features, and size of the TLC.

An event may have its triggering events and follow-up events. For example, a murder case may be the triggering event for the event of arresting a suspect after a while. The latter may have a follow-up event of conviction of the suspect. Such events may share much common content. The temporal locality of TLCs helps distinguish the documents related to an event from those related to its triggering and follow-up events. It significantly distinguishes our approach from existing clustering-based text collection visualization approaches that only consider content coherency. The latter often mix the documents about an event and its triggering and follow-up events in the same cluster and do not allow users to observe the temporal features of the individual events.

To allow users to detect big stories consisting of related events and track event evolutions, **TLC groups** are automatically constructed. A TLC group consists of TLCs with closely related content, no matter when the documents are released. It maps to a big story consisting of the real life events associated with its member TLCs. The events can be contemporary and reflect different aspects of the story. They can also be sequential (triggering events and follow-up events) and reveal how the story develops. Thus a TLC group allows users to examine the *evolution of events* in a large temporal and semantics context.

Besides revealing events, big stories, and their evolution, organizing a large TCTF into TLCs and TLC groups also enables multi-resolution visual exploration. In particular, users can explore the TCTF at three levels of abstraction, namely documents, TLCs, and TLC groups, in EventRiver. It is important to note that the number of documents is much larger than that of TLCs and TLC groups. For example, in the text collection shown in Fig. 1, $29,211$ documents are aggregated into 363 TLCs and 17 TLC groups. Therefore, TLCs and TLC groups provide a compact, contextual representation of all the information at comprehensible and meaningful levels of abstraction. Meanwhile, users can zoom into TLC groups of interest for detailed understanding of their narrative arcs. They can also inspect a TLC at the individual docu-



Wed Aug 16 2006

1. THERE'S BEEN AN ARREST IN CONNECTION WITH THE MURDER OF JONBENET RAMSEY NEARLY TEN YEARS AGO AT HER HOME IN BOULDER, COLORADO.
2. A SUSPECT HAS BEEN PICKED UP AND ARRESTED IN BANGKOK, THAILAND.
3. Reporter: THE MAN THAT WAS ARRESTED IN BANGKOK, THAILAND IS A MIDDLE AGED AMERICAN.
4. AND WOLF, WE DID GET A CONFIRMATION, AN OFFICIAL CONFIRMATION, FROM ICE, THAT SAYS THAT IT'S PLEASED TO ASSIST THE

Thu Aug 17 2006

1. HAPPENING THIS MORNING IN THAILAND, A CONFESSION IN THE 1996 KILLING OF JONBENET RAMSEY.
2. 41-YEAR-OLD JOHN MARK KARR TELLS REPORTERS HE WAS WITH THE GIRL WHEN SHE DIED.
3. KARR CLAIMS THE DEATH WAS AN ACCIDENT.
4. THE PUBLIC HAS LONG BEEN FASCINATED WITH THE DETAILS OF THE JONBENET MURDER FOR ALMOST TEN YEARS.
5. THE BEAUTY PAGEANT STAR WAS STRANGE-WEEK-OLD A PIECE OF SHORT

Fri Aug 18 2006

1. WE REVEAL THE E-MAILS THAT JOHN KARR APPARENTLY WROTE TO PROFESSOR MICHAEL TRACEY AT THE UNIVERSITY OF COLORADO.
2. FOR MORE ON THESE REMARKABLY CREEPY MESSAGES WE HEAD BACK TO BOULDER, COLORADO, AND MY COLLEAGUE TOM FOREMAN.
3. IT IS FASCINATING TO THINK THAT THESE ARE THE THOUGHTS OF THE MAN WHO KILLED JONBENET.
4. THE E-MAILS PAINT A DISTURBING PICTURE OF JOHN KARR AFTER CALIFORNIA POLICE CHARGED HIM WITH

Sun Aug 20 2006

1. IT IS SUNDAY, AUGUST 20th, 9:00 A.M. IN ATLANTA, JOHN MARC KARR HAS HEADED TO THE UNITED STATES.
2. NEW DEVELOPMENTS UNFOLDS THIS HOUR IN THE JONBENET RAMSEY CASE.
3. RIGHT NOW SUSPECT JOHN MARK KARR IS ON A PLANE THAT WILL BRING HIM BACK TO THE UNITED STATES TO FACE CHARGES.
4. KARR IS DUE TO ARRIVE IN LOS ANGELES TONIGHT.
5. CNN'S STAN GRANT WAS AT THE AIRPORT IN BANGKOK, THAILAND, WHEN KARR ARRIVED.

Tue Aug 22 2006

1. ABOUT THE ONLY THING NEW WE KNOW FOR SURE RIGHT NOW IN THE JONBENET RAMSEY CASE, JOHN MARK KARR IS HEADED FOR BOULDER, COLORADO.
2. CNN'S DAN SIMON WAS IN THE LOS ANGELES COURTROOM FOR KARR'S EXTRADITION HEARING.
3. JOHN KARR, TO ME, LOOKED TO BE STOIC.
4. WHICH MEANS AT THIS POINT, HE COULD BE TRANSFERRED TO BOULDER, COLORADO, AT ANY POINT.
5. ESSENTIALLY, BOULDER DETECTIVES NEED TO PICK HIM UP AND TAKE HIM.

Thu Aug 24 2006

1. SOME NEW DEVELOPMENTS IN THE JONBENET MURDER CASE.
2. CALIFORNIA ATTORNEYS JAMIE HARMON AND PATIENCE VANZANDT TOLD REPORTERS LATE YESTERDAY THAT JOHN MARK KARR HAS RETAINED THEM.
3. ALSO, THERE'S NEW INFORMATION OUT ABOUT KARR'S PAST.
4. LET'S GET RIGHT TO PETER VILES IN L.A., WHERE KARR IS BEING HELD.
5. THIS IS JOHN MARK KARR'S FOURTH NIGHT IN THE JAIL BEHIND ME, THE LOS ANGELES COUNTY JAIL.
6. THAT SAID, JAMIE HARMON, THE

Fig. 3. A *storyboard* for TLCs about the *Ramsey Murder Case*.

ment level while still keeping contact with the contextual overview. This scalability and meaningful hierarchy of abstraction are key attributes of the event-based analysis presented in this paper.

The automatic analyses in EventRiver are focused on TLC and TLC group construction and characterization. First, the newly arrived documents in a TCTF are characterized into keyword vectors. An existing algorithm is used to conduct this step in a streaming manner (refer to [20] for details of this algorithm). Then, the characterized documents are clustered into TLCs using a novel streaming text clustering algorithm. Finally, the semantic content and strength over time of the TLCs are characterized and the TLCs are organized into TLC groups.

Note that a streaming data model is used in the automatic analyses, namely that both the document characterization and the TLC construction algorithms process the TCTF assuming that its documents are arriving constantly at a high-speed rate. Newly arrived documents are characterized and merged into TLCs without reprocessing existing documents. Thus EventRiver has the potential to scale to fast evolving text streams. In the following sections, we first present the TLC construction algorithm, and then describe the TLC characterization and group construction approaches.

3.2 TLC Construction

We propose an incremental streaming text clustering algorithm named **Temporal-Locality Clustering** to discover TLCs from TCTFs. Following the two-phase data stream clustering paradigm proposed by Aggarwal et al. [21], Temporal-Locality Clustering algorithm consists of an online component, named **Parallel Processing Component (PPC)**, to periodically store detailed summary statistics of newly arrived raw data, and an offline component, named **Sequential Processing Component (SPC)**, to use the summary statistics generated by PPC for higher level data aggregation without processing the raw data.

In particular, the time horizon of a text stream is divided into a sequence of non-overlapping **Short Time Horizons (STHs)** with an equal length W_{STH}. They are sorted in chronological order and denoted by $STH_i (i = 0, 1, 2, \ldots)$. A document whose releasing time falls into STH_i is called a document in STH_i. The set of all documents in STH_i is denoted by $D(STH_i)$. Our assumption is that the time differences among $D(STH_i)$ are trivial to the application. Therefore, W_{STH} is application-related. For example, we set W_{STH} to be 24 hours when we examine news documents in several months.

PPC processes the TCTF with a single scan. It clusters $D(STH_i)(i = 0, 1, \ldots)$ one by one in the chronological order of the STHs. SPC is in charge of maintaining the TLCs. Whenever PPC finishes clustering $D(STH_i)$, SPC updates the TLCs by merging the clustering results into current TLCs or creating new TLCs. PPC and SPC are described as follows.

3.2.1 Parallel Processing Component

PPC processes all documents in one STH in each run, ignoring their time differences and forgetting other documents in the TCTF. The documents are represented by their keyword vectors generated by the algorithm presented in [20].

PPC clusters the documents into groups named **Intermediate Clusters (ICs)**. Documents within the same IC have coherent content and documents in different ICs have distinct content. Each IC records the IDs and the keyword vectors of its member documents. The clustering algorithm used by PPC is Rock [22], a robust hierarchical clustering algorithm for data with Boolean and categorical attributes. It uses Jaccard Coefficient to define the neighborhood among the data items. The similarity between two data items is measured by the number of their common neighbors. We chose Rock since it generates high quality clusters and scales to large datasets (its worst-case time complexity is $O(n^2 + nm_m m_a + n^2 logn)$, where n is the number of input data items and m_m and m_a are the maximum and average numbers of neighbors for all data items) [22].

3.2.2 Sequential Processing Component

SPC creates and maintains TLCs. Once PPC finishes processing $D(STH_i)$, the resulting ICs, namely the document clusters detected from $D(STH_i)$, are sent to SPC. According to the content coherence and temporal locality criteria, SPC either merges an IC into an existing TLC or create a new TLC for it.

Let $tc1$ and $tc2$ be the starting time and ending time of STH_i and q be the set of ICs detected from $D(STH_i)$. Let Q be the set of all TLCs detected from the TCTF before q is processed (Q is empty before STH_0 is processed). Let tl be the time when a TLC was last updated. We define the **Active TLC Set** $Q^a = \{TLC_i | i = 0, 1, 2, \ldots, TLC_i \in Q and (tc1 - tl(TLC_i)) < t_\delta\}$, where t_δ is a threshold to ensure temporal locality (t_δ is usually set to be CW_{STH}, where C is a constant larger than 1 and W_{STH} is the width of a STH). ICs can only be merged into TLCs in Q^a, since these TLCs were updated recently and thus they maintain temporal locality. The pseudo code of SPC is shown in Fig. 4. The resulting TLCs are retrieved from Q.

The time complexity of SPC for processing an STH is $O(RP^2)$, where P and R are the sizes of q and Q^a respectively. To fasten the calculation of Jaccard Coefficients between an IC and a TLC, samples of the documents in them can be used rather than all the documents.

3.3 TLC Characterization

We characterize a TLC using its *Strength over time* and *semantic representation*. They are described in the following sections.

Fig. 4. SPC Pseudo code

```
Begin $SPC(q, Q, Q^a, tc1, tc2, t_\delta)$
for each $TLC_i \in Q^a$ do
    for each $IC_j \in q$ do
        $Sim_{i,j}$ := Jaccard Coefficient between $TLC_i$ and $IC_j$;
    end for
end for
for each $TLC_k \in Q^a$ do
    Get $j_{max}$ where $Sim_{k,j_{max}} = \max\{Sim_{k,j} | \forall TLC_j \in Q^a\}$;
    if $Sim_{k,j_{max}} = \max\{Sim_{i,j_{max}} | \forall IC_i \in q\}$ then
        $Merge(IC_{j_{max}}, TLC_k)$;
        $tl(TLC_k) = tc2$;
        $Remove(IC_{j_{max}}, q)$;
    end if
end for
for all $TLC_i \in Q^a$ do
    if $tc2 - tl(TLC_i) \geq t_\delta$ then
        $Remove(TLC_i, Q^a)$;
    end if
end for
for all $IC_i \in q$ do
    Make $IC_i$ a new $TLC_m, m \in N$;
    $Insert(TLC_m, Q^a)$;
    $Insert(TLC_m, Q)$;
end for
End $SPC(q, Q, Q^a, tc1, tc2, t_\delta)$

Begin $Merge(IC_{j_{max}}, TLC_k)$
for each document $d_x \in IC_{j_{max}}$ do
    Add $d_x$ into $TLC_k$
end for
End $Merge(IC_{j_{max}}, TLC_k)$
```

3.3.1 Strength over time

The **Strength over time** of a TLC is defined as:

***Definition** 1:*

$$S(TLC, t) = Count(TLC, STH(t)) \tag{1}$$

where $STH(t)$ is the STH that covers time t, and $Count(TLC, STH(t))$ returns the number of documents in TLC whose releasing time fall in $STH(t)$.

$S(TLC, t)$ reveals how the motivating event of TLC receives continuous attention by counting the number of documents in TLC over time. The underlying assumption is that the more important an event is, the more documents will be published to report and discuss it.

3.3.2 Semantic Representation

Since the keyword vectors of the documents in a TLC often involve a large number of keywords, it is impossible to use all of them to depict the semantics of the TLC. Thus, we propose a novel dual labeling approach to depict the semantics.

***Definition** 2:* The **Intra-Link Co-strength** $ILC(k_i, k_j)$ between keyword k_i and k_j in a TLC is the number of its documents in which k_i and k_j co-occur.

***Definition** 3:* The **Intra-Link Significance** of keyword k_i in a TLC is defined as $ILS(k_i) = \sum_{k_h \in U, k_h \neq k_i} ILC(k_i, k_h)$, where U is the union of all keywords used to characterize any documents in the TLC.

To depict the semantics of a TLC, the ILS of each keyword in U is calculated. The keywords with the highest ILSs and the lowest ILSs are used to represent the semantics of the TLC. The keywords with the highest ILSs are the most shared keywords in the TLC and describe its semantic context. They are named **Context-Keywords** of the TLC. The keywords with the lowest ILSs convey unique contents in the documents and named **Core-Keywords** of the TLC. Assuming that the distribution of keywords is normal, we set the **boundary** b_e for context-keywords and core-keywords as $\mu - \sigma$, where μ and σ are the mean and standard deviation of the ILSs, respectively. The semantic representation of the TLC is the combination of the **context-keywords** and the **core-keywords**, and thus called **dual labeling**.

3.4 TLC Group Construction

TLCs are clustered into **TLC groups** using an algorithm similar to the SPC algorithm, where TLCs replace ICs and TLC groups replace TLCs. The similarity between TLCs is calculated based on their context-keywords using Jaccard Coefficient. The event groups are also updated after each STH is processed.

A TLC is considered an **outlier** if it meets one of the following criteria: (1) it is the only member of its TLC group, namely that its motivating event has no related events; (2) there are other TLCs in its TLC group, but all the TLCs in this group only occupy the same single STH. It means that the motivating event and its related events do not receive continuous attention. In the rest of this paper, TLC groups refer to the groups without outlier TLCs.

4 Visualization and Interactions

EventRiver visually presents the automatic analysis results to users and allows them to interactively conduct the ERTs without prior knowledge of the TCTF. Its visualizations and interactions are described in detail in the following sections.

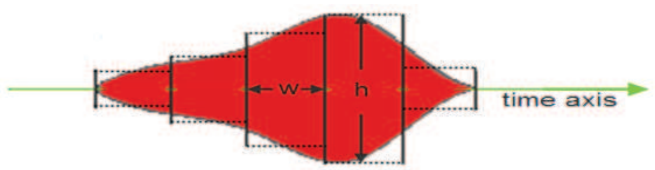


Fig. 5. The bubble representation of a TLC

4.1 Visualization

EventRiver displays a TCTF as a river of bubbles flowing over a horizontal time axis (see Fig. 1 for an example). We first introduce the bubble metaphor, second describe how the bubbles are laid out in the river to form the temporal context with big stories distinguished, and then present how the semantics of the TLCs and TLC groups are displayed.

4.1.1 Bubbles

EventRiver uses a bubble to represent a TLC, as shown in Fig. 5. Its horizontal width represents the life span of the TLC. Its vertical width approximates the strength of the TLC over time $S(TLC, t)$ (refer to Section 3.3). By comparing the shapes of different bubbles, users can find events that receive most significant attention since the bubbles of their TLCs are bigger and longer than the other bubbles.

The curved shape of a bubble is a smoothed approximation of a set of rectangles as illustrated in Fig. 5. The rectangles depict $S(TLC, t)$. Each rectangle corresponds to a STH in which there are documents in this TLC. Its horizontal width (W) is W_{STH} and its vertical width (h) represents the number of documents in the TLC that fall into this STH. The curve boundaries of the bubbles are the Cubic Spline Interpolation of these rectangles.

The color of a bubble is assigned based on the TLC group to which it belongs. TLCs in the same group are in the same color. All outlier (refer to 3.4) TLCs are colored in dark grey to reduce color clutter on the screen.

4.1.2 Bubble Layout

The bubbles are laid out in the river to form a temporal context. TLC groups are also distinguished by the bubble layout. In addition, the number of TLCs can be large and thus the bubble layout needs to be carefully designed to reduce clutter.

The bubbles are positioned along the time axis according to the STHs the TLCs cover. Users can observe when the TLCs begin, burst, and fade according to their horizontal positions. The vertical positions of the bubbles convey the relationships among the TLCs. Four rules are followed in the vertical position assignment: (1) TLCs belong to the same group should be adjacent in their vertical positions to visually reveal their close relationships; (2) More important TLC groups should be higher in their vertical positions than less important TLC groups since the reading manner of human beings is usually from top to bottom. The **importance of TLC groups** can be defined using different criteria, such as the number of documents included, the duration time, the starting time, the ending time, or the maximum number of documents involved in any individual STH. Users can interactively set the criterion; (3) Within a TLC group, TLCs are settled one by one in the highest possible vertical positions. The priority of TLC settling from high to low is starting time, peak strength, and duration; (4) No overlaps are allowed among the bubbles.

The vertical positioning algorithm is described as follows: **Step 1:** Place the TLC groups one by one in descending order by the group importance. When placing a group, the highest vertical position can be assigned to its member TLCs is the topmost vertical position in the group time horizon below which there are no other bubbles placed. There is no constrain to the lowest vertical position, namely that the vertical space is assumed

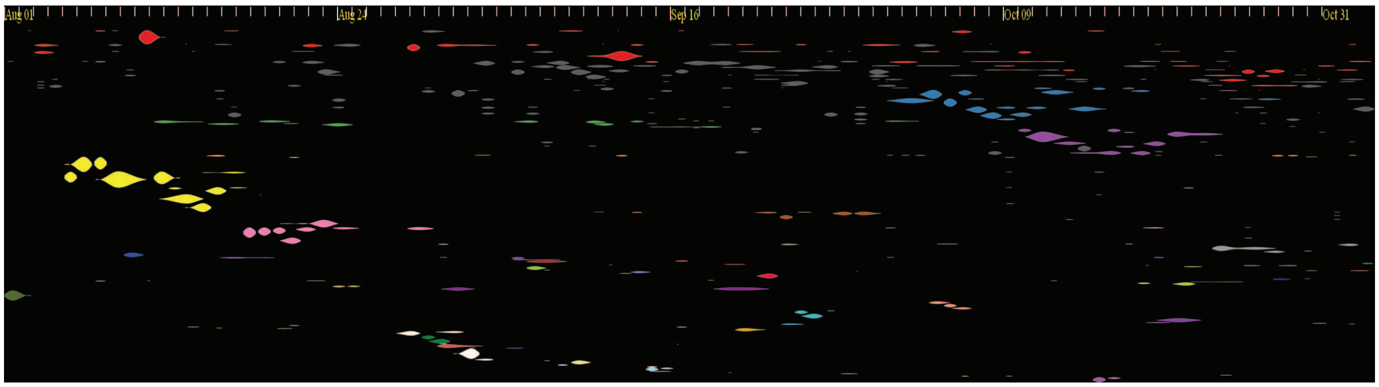


Fig. 6. Bubble layout with 1901 TLCs. The TCTF displayed is CNN news from Aug. 1 to Oct 31, 2006 (72,435 closed-caption documents).

to be unlimited. Thus the group is placed within the half-open vertical range defined by the highest vertical position, called the vertical space of the group. The TLCs of the group are placed one by one according to their priorities. Each TLC is placed in the highest possible vertical position within the vertical space of the group without overlapping the TLCs settled. **Step 2:** Insert the outlier TLCs to the topmost unoccupied space one by one according to the same priority list used in settling TLCs in the same group. **Step 3:** Scale the positions and the vertical sizes of all the TLCs to fit them into the screen space. The relative vertical sizes of the TLCs are kept so that users can compare the strengths of multiple TLCs.

Fig. 6 gives an example of the bubble layout. The TCTF displayed is all CNN news in duration of three months. All the 1901 TLCs extracted are displayed. Several big TLC groups, such as the groups in yellow, in pink, in blue, and in red, can be noticed from the display. Users can examine these groups for major stories driving this TCTF.

4.1.3 Semantic Representation

The semantics of the TLCs are dynamically displayed as labels and snippets of the bubbles (see Fig. 7 for an example). A bubble is labeled by **dual-labels**, where core-keywords and context-keywords (refer to Section 3.3) are displayed in parallel and in distinct background colors. It is interesting that TLCs in the same group often have shared context-keywords and unique core-keywords. As a consequence, dual labels help users learn the common theme in the TLC group and detect the unique content of the individual TLCs. For example, Fig. 7 shows TLCs about the 2006 Lebanon War with their labels displayed. The labels in white background provide context information while the labels in yellow background display the unique content of each TLC. They allow users to build a rough mental map of the long-running stories without reading the documents. Note that the yellow labels in this example contain a few unrelated keywords which are introduced by errors in the segmentation of the closed caption documents.

To reduce clutter, EventRiver provides a few automatic labeling strategies to selectively label TLCs. **Representative TLC labeling** automatically labels a representative TLC, such as the TLC with the largest number of documents, for each TLC group. **Outlier labeling** labels outlier TLCs only. In addition, when users zoom into a TLC group to examine it, all of its TLCs will be automatically labeled.

Users can click on a TLC to manually turn on/off its labels. In the case that the labels of several TLCs overlap, users can manually relocate a TLC and its labels by mouse drag and drop. TLCs and their labels can only be moved vertically since their horizontal positions indicate their temporal attributes and should not be modified.

When the labels do not fulfill the information need of a user, the user can gain more semantic information about a TLC using **on-the-fly snippets**. In particular, holding a button of the mouse on a TLC will trigger a snippet showing a few sentences from the documents in the TLC that contain one or more core-keywords and context-keywords. Fig. 7 gives an example of the snippet.

4.2 Interactions

The following interaction tools are provided in EventRiver. They are categorized according to the ERTs they support:

Interactions for Event Browsing:

- **Filtering-by-Strength** - Users can set a strength range to hide the TLCs whose peak strengths are out of the range from the screen. By hiding TLCs with small strengths, users can examine major events driving the text without the distraction of the small ones, as shown in Fig. 1. By hiding TLCs with large strengths, more room can be given to less significant events.
- **Semantic Zooming** - Users can use semantic zooming to remove unselected TLCs from the screen and rescale the selected TLCs to fulfill the screen. Meanwhile, the labels of the selected TLCs will be automatically turned on.

- **Temporal Zooming** - Users can brush the time axis to select a specific time period as a focus time slot. This time slot will be rescaled to fulfill the horizontal screen space.
- **Group Sorting** - Users can sort the TLC groups by a different importance criterion (refer to Section 4.1.2) and regenerate the vertical layout so that TLC groups considered more important will be displayed higher in the display.
- **Manual Relocation** - Users can manually change the vertical positions of individual TLCs to reduce overlaps among the labels.

Interactions for Event Search, Tracking, and Association:

- **Search-by-Keywords** - Users can search TLCs that contain/do not contain documents with user-defined keywords in their keyword vectors. Users can type the keywords or select them from a keyword list in the search box (see Fig. 2(a) for an example). The list contains all the keywords used to characterize documents in the TCTF and can be sorted by different criteria, such as the total occurrences, the peak counts within the STHs, and the first/last appearance time.
- **Search-by-Text** - Users can search TLCs of interest by a piece of example text. All the TLCs containing documents that share characterizing keywords with the example text will be selected.
- **Search-by-TLC** - Users can interactively select a TLC of interest as a focus and search all the TLCs that share any core-keywords or context keywords with it. This interaction allows users to find associated TLCs and track the event evolution.
- **Storyboard** - Users can open a Storyboard (Fig. 2(b)) for the search results of the above interactions. In the storyboard, there is a shoebox (please refer to *Interactions for Event Investigation* for more details) for each TLC selected. The shoeboxes are placed side by side in the order of the starting times of the TLCs so that users can examine associated events in chronological order to construct a metal map of the whole story. Fig. 3 and Fig. 2(b) give examples of a shoebox and a storyboard respectively.

In addition, users can use semantic zooming to examine the selected TLCs in a less cluttered river.

Interactions for Event Investigation:

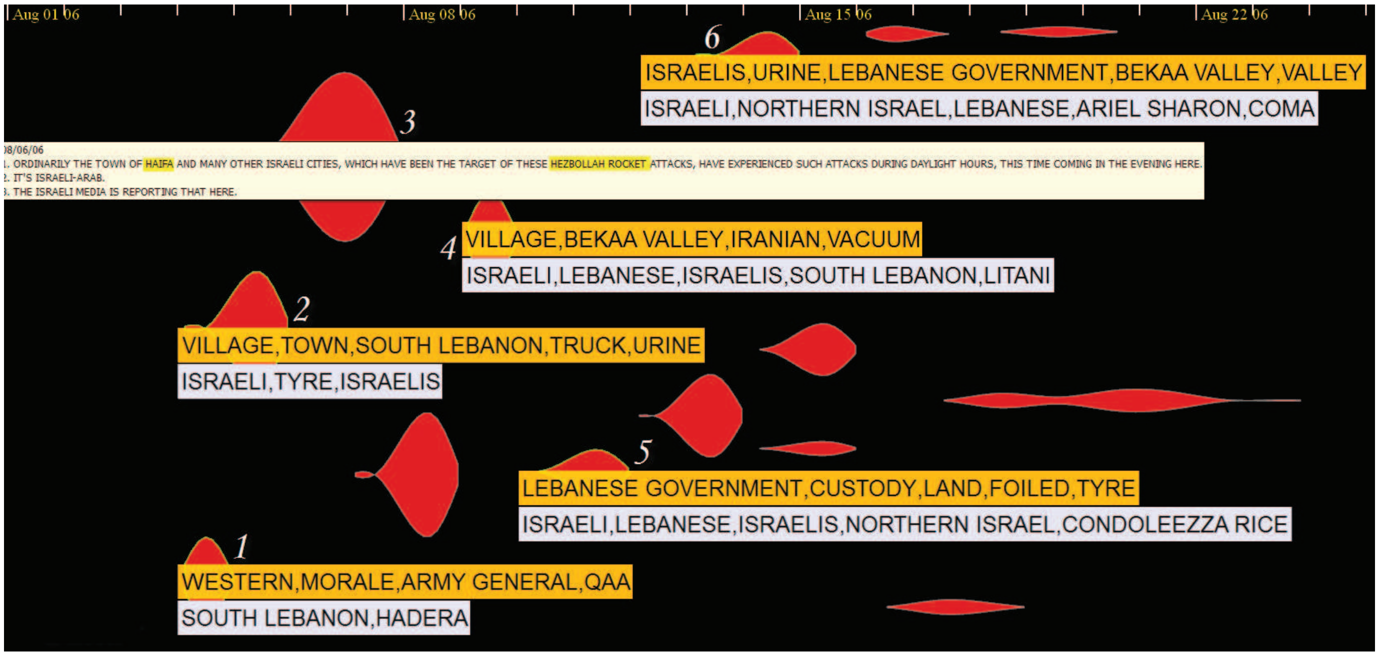
- **Shoebox** - Users can click on a TLC to open a shoebox (see Fig. 2(b) for an example) for examining its documents in detail and conducting investigative analysis. Its interface consists of three components:
  - **Index Panel** lists all documents within the TLC, which can be sorted by their lengths, releasing time, or other criteria. Typing a keyword in the search box will remove all documents with no occurrence of the keyword from the list. Clicking a document in the list will load the document to the detail panel.
  - **Detail Panel** allows users to read a document in snippets or full text. In the snippet mode, only the sentences containing core-keywords or context-keywords are displayed. In both modes, the keywords are highlighted by colors. For the news video application, the video associated with the closed-caption document can be played in a playback window.
  - **Evidence Box** allows users to save documents of interest into evidence files for external uses, such as evidence exchange or hypothesis evaluations.

5 Case Study

In this section, we present a few case studies. They show how EventRiver help users quickly browse a large TCTF for major driving events, track their evolution, and dive in an event for investigative analyses. The data explored is CNN news from Aug. 1 to Aug. 24, 2006, containing $29,211$ closed-caption documents.

Case Study 1: Event Browsing - In this case, we quickly browse the major real life events and long-running stories that drive the news without any prior knowledge. When the dataset is opened in EventRiver, all TLCs detected from it are displayed. We interactively *filter* out TLCs with small strengths and *label* the representative TLCs of significant TLC groups. The resulting display is shown in Fig. 1. Several big TLC groups stand out. The red TLC group contains context-keywords “Israeli” and “Lebanese”. From the snippets we learn that this group is about the **2006 Lebanon War**. The blue TLC group has the context-keywords “Jonbenet Ramsey” and “Thailand”. From the snippets we learn that it reports the new leads in the **Jonbenet Ramsey Murder Case**. Similarly, several other big stories, such as the **2006 Transatlantic Aircraft Plot** (corresponding to the green TLC group) and **Floyd Landis Drug Scandal of La Tour de France** (corresponding to the pink TLC group), are detected.

Case Study 2: Event Search, Tracking, and Association - In this case, we take a closer look at the **Jonbenet Ramsey Murder Case**. We select all TLCs about this story using the *Search-by-TLC* interaction and open a *Storyboard* for them (Fig. 3). In Fig. 3, the snippets in shoeboxes exhibit the narratives of this story, starting with an motivating event on *Aug.16* - “There’s been an arrest in connection with the murder of Jonbenet Ramsey nearly ten years ago at her home in Boulder, Colorado”. Then there were the follow-ups of this event: *Aug.17*, Karr admitted being in the company with Jonbenet when she was killed; the next day, *Aug.18*, an e-mail written by Karr was revealed as a piece of importance evidence in this murder case; on *Aug.20*, suspect John Karr was deported to the United States from Thailand for trial; Karr waived extradition in Los Angeles County Superior Court, clearing the way for his transfer to Boulder (CO) on *Aug.22*; and *Aug.24*, Karr retained his California attorneys for the upcoming trial in Boulder (CO).


Aug 01 06
Aug 08 06
Aug 15 06
Aug 22 06
6
ISRAELIS,URINE,LEBANESE GOVERNMENT,BEKAA VALLEY,VALLEY
ISRAELI,NORTHERN ISRAEL,LEBANESE,ARIEL SHARON,COMA
3
4
VILLAGE,BEKAA VALLEY,IRANIAN,VACUUM
ISRAELI,LEBANESE,ISRAELIS,SOUTH LEBANON,LITANI
2
VILLAGE,TOWN,SOUTH LEBANON,TRUCK,URINE
ISRAELI,TYRE,ISRAELIS
5
LEBANESE GOVERNMENT,CUSTODY,LAND,FOILED,TYRE
ISRAELI,LEBANESE,ISRAELIS,NORTHERN ISRAEL,CONDOLEEZZA RICE
1
WESTERN,MORALE,ARMY GENERAL,QAA
SOUTH LEBANON,HADERA

)8/06/06
. ORDINARILY THE TOWN OF HAIFA AND MANY OTHER ISRAELI CITIES, WHICH HAVE BEEN THE TARGET OF THESE HEZBOLLAH ROCKET ATTACKS, HAVE EXPERIENCED SUCH ATTACKS DURING DAYLIGHT HOURS, THIS TIME COMING IN THE EVENING HERE.
. IT'S ISRAELI-ARAB.
. THE ISRAELI MEDIA IS REPORTING THAT HERE.

Fig. 7. Zooming into the TLC group about the *2006 Lebanon War*. The on-the-fly snippet of TLC 3 is displayed.

Case Study 3: Event Investigation - When we zoom in the **2006 Transatlantic Aircraft Plot** TLC group (see Fig. 2(a)), we notice that the keyword "gel" appears in the labels of a few TLCs. We wonder how *gel* is related to this terrorism and click on a TLC with "gel" in its label to open a shoebox (see Fig. 2(b)) to investigate it. In the shoebox, we search documents containing "gel" and learn that "gel" has been listed as a suspicious and banned item in air flights to prevent the carrying of disguised liquid explosives.

6 Experiment

The effectiveness of EventRiver in supporting ERTs heavily depends on the quality of the TLCs extracted by the clustering algorithm. Its scalability to evolving text collections is also related to the time efficiency of this algorithm. We have conducted a set of experiments to evaluate the efficiency and effectiveness of this algorithm.

6.1 Settings

A real dataset, namely the closed-caption stream of CNN news videos from Aug. 1 to 24, 2006 was used in the experiment. The closed-caption stream is segmented into $29,211$ text documents to form a TCTF. On average 9092 keywords per day were used to characterize the documents.

EventRiver processed the TCTF on a PC with Intel Core 2 Duo processor and $2GB$ memory. The parameters of the algorithms were set as follows: the length of STH $W_{STH} = 24h$; A sampling rate of 10% was used in SPC Jaccard Coefficient calculation for ICs and TLCs whose sizes are larger than 10. Otherwise, the sampling rate is 100%

6.2 Time Efficiency

The time of TLC extraction and TLC group construction for the whole TCTF was $5m13s$, which means that it takes about $0.01s$ to process each document on average. In particular, the average processing time for one STH (there are 1221 documents per STH on average), namely the average incremental processing time, was $13.064s$. The result showed that the TLC extraction and TLC group construction were time efficient.

6.3 Quality

There were 363 TLCs and 17 TLC groups extracted from the TCTF (see Fig. 1). A multi-resolution approach was used to examine the results. First, to make sure that no major stories in the TCTF were missed in EventRiver, we visualized the same TCTF using IN-SPIRE [6] (see Fig. 8) and compared the TLC groups with the clusters revealed in IN-SPIRE. IN-SPIRE clustered the whole TCTF without considering the time stamps of the documents and displayed the clusters as mountains. As shown in Fig. 8, IN-SPIRE revealed 18 significant clusters excluding the identical/similar ones. Comparing them with the 17 TLC groups from EventRiver, we found that the two systems detected 15 topics in common. The two exclusive topics from EventRiver were "Floyd Landis drug scandal of La Tour de France" and "the kidnap of the two journalists Olaf Wiig and Steve Centanni in Gaza". By scanning the original dataset, we were sure that both of them were significant topics. The three exclusive topics from IN-SPIRE were labeled as "larry, ve, didn't", "hot, companies, women", and "larry, ve, ricky, year". After reading documents in these topics, we learned that "larry" refers to the famous *CNN* anchor *Larry King* and the topics

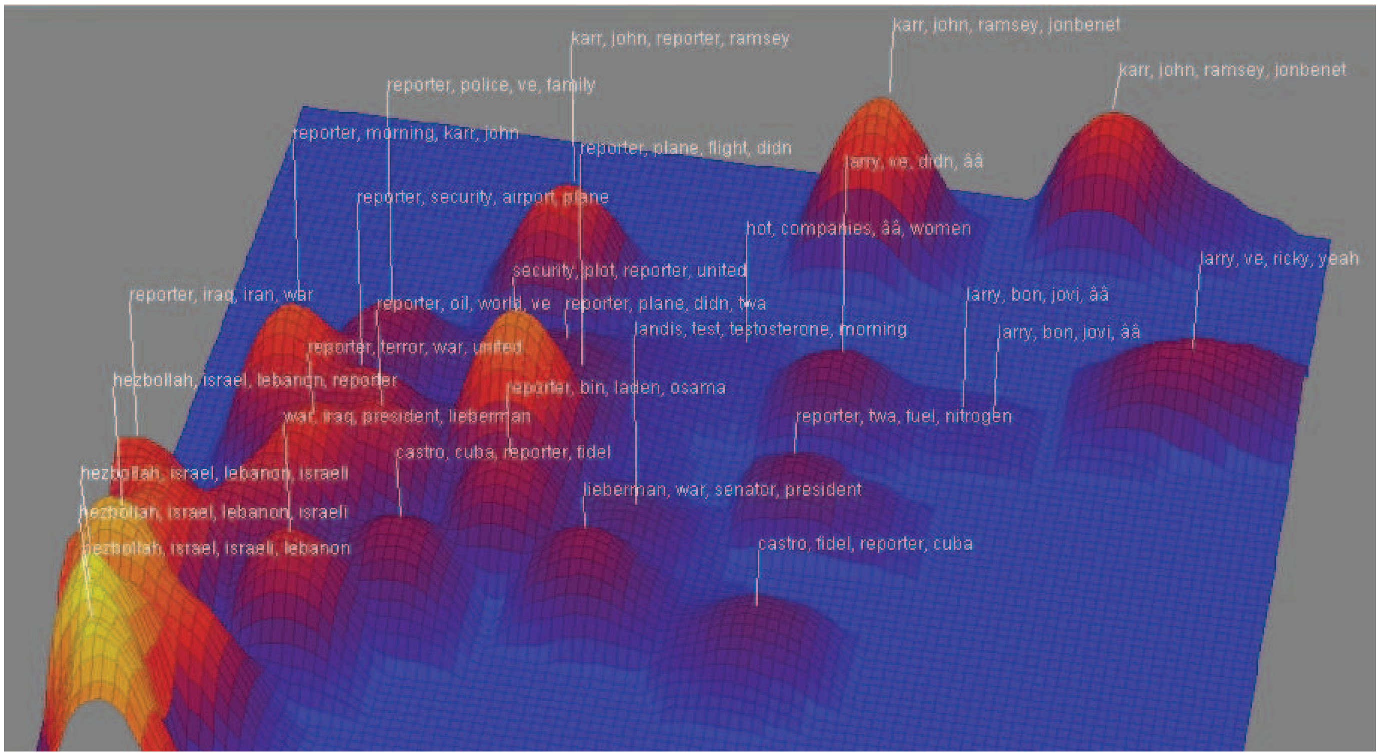

karr, john, reporter, ramsey
karr, john, ramsey, jonbenet
karr, john, ramsey, jonbenet
reporter, police, ve, family
reporter, morning, karr, john
reporter, plane, flight, didn
larry, ve, didn, ââ
reporter, security, airport, plane
hot, companies, ââ, women
security, plot, reporter, united
larry, ve, ricky, yeah
reporter, iraq, iran, war
reporter, oil, world, ve
reporter, plane, didn, twa
larry, bon, jovi, ââ
landis, test, testosterone, morning
larry, bon, jovi, ââ
reporter, terror, war, united
hezbollah, israel, lebanon, reporter
reporter, bin, laden, osama
war, iraq, president, lieberman
reporter, twa, fuel, nitrogen
castro, cuba, reporter, fidel
hezbollah, israel, lebanon, israeli
hezbollah, israel, lebanon, israeli
lieberman, war, senator, president
hezbollah, israel, israeli, lebanon
castro, fidel, reporter, cuba

Fig. 8. *CNN* news from Aug. 1 to 24, 2006 ($29,211$ closed-caption documents) in IN-SPIRE.

consist of several small events in Larry King's show. The topic "hot, companies, women" consists of a set of trivial daily news with these three words appearing prominently.

The results showed that EventRiver captured the major stories driving the TCTF as effectively as or even better than IN-SPIRE. EventRiver also showed significant advantages in visually depicting the temporal evolution of the stories in one view and explicitly revealing their key events. IN-SPIRE depended on time slicing techniques to reveal such information, which was not efficient and suffered from change blindness [7].

We next examined several long-running stories detected by EventRiver in detail by comparing their TLCs with an external reference source, namely **Wikipedia.org** [23]. Eight stories with the largest numbers of events were examined, each of which was listed as one of the weekly top 5 most popular topics at CNN.com at least once for the time period during which it occurred. In Wikipedia, the major events of these stories were manually summarized and listed as timelines. We believe that such human-generated summaries are usually more accurate than any automatic results, which make them a benchmark for evaluation.

For each of the eight stories, we first manually compared its TLCs with the events listed in the Wikipedia timeline to identify matching events. Then, we measured the quality of EventRiver results against Wikipedia using *precision* [24], namely the ratio between the number of matching events and the total number of TLCs in the TLC group, and *recall* [24], namely the ratio between the number of matching events and the total number of events listed in the Wikipedia timeline. This process is illustrated in the following example (refer to Appendix for more examples).

The 2006 Lebanon War - A Close Look at the Event Extraction Results

Wikipedia has a timeline [23] for the military operations of the 2006 Lebanon War where 17 events are listed in the time period of the experiment dataset. For EventRiver, there were 15 TLCs in the TLC group of the 2006 Lebanon War. 14 events matched, which led to a precision of $93.3\%(14/15)$ and a recall of $82.4\%(14/17)$.

Here we list a few example matching events. For each event, we cite the event description in Wikipedia and highlight the keywords appearing in the labels of the matching TLC in EventRiver in bold (see Fig. 7 for the TLCs, labels, and snippets in EventRiver): (1) On *Aug.4*, "Israel targeted the **southern** outskirts of Beirut, and later in the day, Hezbollah launched rockets at the **Hadera** region. 33 civilian farm workers are killed and 20 wounded after an Israeli airstrike in a farm near **Qaa** in **Lebanon**." (2) On *Aug.5*, "**Israeli** commando soldiers landed in **Tyre**, where fighting erupted with Hezbollah forces." (3) On *Aug.6*, "12 army reservists resting near the Lebanon border were killed in the deadliest barrage of **Hezbollah rocket** attacks so far. Three **Israeli** civilians were also killed in a dusk attack in the port of **Haifa**." A snippet of this TLC is shown in Fig. 7. (4) On *Aug.9*, "in the eastern **Bekaa Valley** five people were reported killed and two feared dead after an **Israeli** airraid." (5)

	Topics (Event Groups)	Precision (%)	Recall (%)
1	Jonbenet Ramsey Murder Case	40.0 (8/20)	72.7 (8/11)
2	2006 Lebanon War	93.3 (14/15)	82.4 (14/17)
3	2006 Transatlantic Aircraft Plot	70.0 (7/10)	77.7 (7/9)
4	Cuban transfer of presidential duties	80.0 (4/5)	80.0 (4/5)
5	Landis' drug use in the 2006 Le tour de France	66.7 (2/3)	100.0 (2/2)
6	Joe Lieberman vs. Ned Lamond in Democratic Party Senate election	75.0 (6/8)	85.7 (6/7)
7	The kidnap of Fox journalists Olaf Wiig and Steve Centanni	66.7 (2/3)	66.7 (2/3)
8	U.N. Security council demands Iran suspend Uranium enrichment	66.7 (2/3)	100.0 (2/2)
	Average	69.1	83.2

TABLE 1

The scores of *precision* and *recall* from the experiment.

On Aug.10, "**Condoleezza Rice** formally explains the resolution plan of the U.N. for reconciliation between Lebanon and **Israel**."; (6) On *Aug.14*, "The Former **Israeli** prime minister **Ariel Sharon**'s health condition gets worse because of a new finding of pneumonia, and he is still in a **coma**."

Table 1 summarizes the precisions and recalls of the stories we examined. Most results were good except for a few exceptions. We therefore looked closer into the exceptions. It turned out that the "Jonbenet Ramsey Murder Case" had a relatively low precision (40.0%, $8/20$) since the Wikipedia authors ignored several events such as interviews with Ramsey's parents and there were some redundant events in EventRiver due to the replay of the same videos several days after their first play. The "Journalist kidnapping" story had a relatively low recall (67.7%, $2/3$) due to one missing event, namely the starting event of this topic that "Steve was kidnapped by Palestinian gunmen while on assignment in Gaza City, along with Olaf Wiig on Aug. 14, 2006". However, we found that the reason was that this news story was exclusively released by FOX News Channel and was not released by CNN. It can be seen from the results that EventRiver successfully provided the main events and the general narrative arc that would result from careful analysis by humans.

The experiment results demonstrate that EventRiver provides an effective and efficient event analysis approach that successfully maps a TCTF to its driving events and provides valid narrative arcs. Since EventRiver effectively organizes a TCTF into semantically meaningful TLCs and TLC groups, the analysis can scale up to much larger TCTFs and can give overviews even when human analyses (demonstrated via Wikipedia here) have not been performed. Further, the temporal analysis approach provided in this paper is unique and gives EventRiver capabilities other tools do not have.

7 User Study

A formal user study was conducted to evaluate the effectiveness and efficiency of EventRiver in supporting the ERTs as a human-centered visual analytics solution.

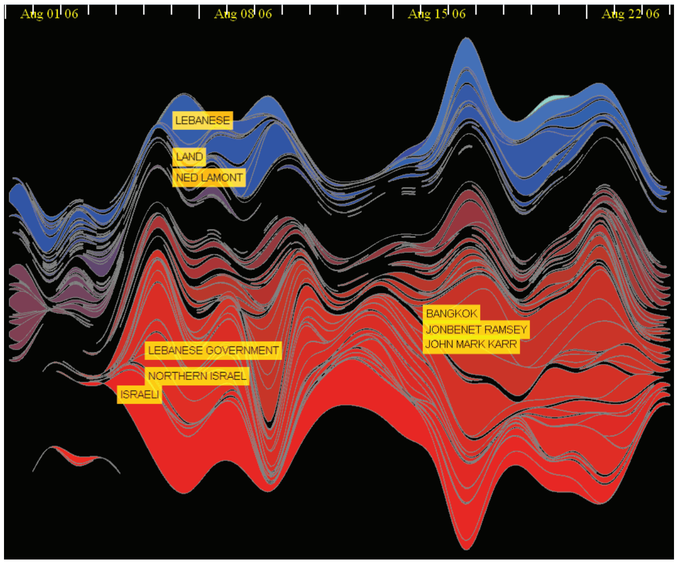


Fig. 9. *CNN* news from Aug. 1 to 24, 2006 ($29,211$ closed-caption documents) in LensRiver. Each current represents a keyword. The x-axis is the time line. The width of a current along the y-axis indicates the strength of the keyword.

We performed this study by comparing EventRiver with LensRiver [12] (see Fig. 9), which is a Keyword-Tracing Technique (KTT) (refer to Section 2). Our assumption was that the EventRiver overview was more effective in helping users discover major events driving a TCTF and learn their semantic contents and evolution than KTTs such as LensRiver.

To form a comparable study, labels and snippets were provided in both EventRiver and LensRiver. The shoebox was disabled to make sure that users learned the semantics without reading the documents. CNN news from Aug. 1 to 24 in 2006 was used in the formal test while CNN news from Sept.1 to 15 in 2006 was used in the training. Each document is characterized by the same keyword vector in both systems. Fig. 1 and Fig. 9 show the test dataset in EventRiver and LensRiver respectively.

The user study was a within-subjects, balanced user study. Twelve graduate students majoring in computer science participated in the user study. All subjects claimed that they were unfamiliar with or forgot about the details about the news during Aug. 2006. The subjects performed the study one by one on the same desktop PC.

The user study was conducted as follows. Each subject worked through two sections. In each section, a subject was asked to complete the same set of tasks with either EventRiver or LensRiver. Half of the subjects worked with EventRiver in the first section and LensRiver in the second section. Another half worked in reversed order. We asked the subjects to use both systems to learn their preference. In our task performance statistics, only the results from the first section were calculated to avoid the case that what a subject learned from the first section helped him/her in the second section. Each section started with a 10 minutes training period in which the

Topic 1.	**Fidel Castro and Cuba Situation**
HE 1	Fidel Castro meets General Abizaid in Cuba.
HE 2	Juanita Castro, sister of Fidel Castro took an interview about Fidel's health condition.
HE 3	Raul Castro takes over the power from Fidel Castro.
Topic 2.	**2006 Transatlantic Aircraft Plot**
HE 1	The Homeland Security secretary Michael Chertoff talked about the London airport plot and explained the new carry-on rules of flights.
HE 2	Some terrorists tried to carry liquid explosives disguised as gel, toothpaste, etc. onto the flights from London Heathrow Airport to US and Canada.
HE 3	Some Pakistani arrested for the London Heathrow airport terrorist plot were originally connected to Osama Bin Laden.

TABLE 2

User Study Task 1 - Hypothetical Event Evaluation

Topic 1.	**The 2006 Lebanon War (Lebanon vs. Israel)**
E 1	The Former Israeli prime minister Ariel Sharon's health condition got worse because of a new finding of pneumonia, and he was still in coma.
E 2	Israel expanded the military operations in southern Lebanon trying to pushing their troops into Latini River.
E 3	Condoleezza Rice formally explains the resolution plan of U.N. for reconciliation between Lebanon and Israel.
E 4	Israel's military action in the Bekaa Valley was a violation of the cease-fire agreement with Lebanon.
Topic 2.	**Jonbenet Ramsey Murder Case (the suspect John Karr)**
E 1	John Karr took the DNA test in Boulder, CO.
E 2	John Karr's arrest in Bangkok, Thailand.
E 3	John Karr was sent to Boulder, CO.

TABLE 3

User Study Task 2 - Event Ordering

instructor introduced the interface and the interactions to the subject and the subject freely explored the tool using the training dataset. The instructor answered questions from the subject during the training.

The formal test followed the training. It contained two tasks. The first task tested how the visualization helped users discover significant events that drive the TCTF and learn their semantics. As shown in Table 2, two groups of hypothetical events were given and the subject was asked to judge whether each hypothetical event was true or false. The subject was asked to rate his/her confidence about the judgment on a 1 (low confidence) to 5 (high confidence) scale. The second task tested how the visualization helped users learn event evolution. As shown in Table 3, two groups of associated real life events were given and the subject was asked to number the chronological order of the events within each group. The subject was again asked to rate his/her confidence about the answers on the same scale. Note that all the events used in the tasks were based on news events that were listed in the weekly top

The total time the subject used to finish each task was manually recorded by the instructor. After the test, the subjects were asked to complete a post-test questionnaire on their preference to EventRiver and LensRiver with regard to their usefulness, ease of use, and awareness of context. Scales of

The results of this study are reported in Table 4. The results showed that EventRiver had advantages over LensRiver for the given tasks with regard to time efficiency, correctness, user confidence, and user preference on usefulness, ease of use, and awareness of context. Users commented that it was easy to find keywords associations and track event evolution using EventRiver. By contrast, users found that it was hard to find keyword associations from LensRiver, especially when the keywords under inspection had a relatively long life span.

		EventRiver	LensRiver
Hypotheses Evaluation	Time (min)	4.35	5.78
	Correctness (%)	62.50	56.25
	Confidence (1-5)	3.13	2.48
Event Ordering	Time (min)	5.8	7.1
	Correctness (%)	75.00	49.75
	Confidence (1-5)	3.75	2.75
Preference	Usefulness (1-5)	4.2	2.4
	Ease of Use (1-5)	3.9	2.8
	Awareness of Context (1-5)	4.3	1.9

TABLE 4

Measures on average of the user study.

8 Conclusion and Future Work

In conclusion, EventRiver significantly advances TCTF analysis in that:

- EventRiver supports ERTs through a novel visual analytics approach. By integrating automatic analysis algorithms with visualizations and interactions, users can effectively and efficiently detect significant real life events driving the TCTFs, track their evolution, and investigate the related documents in full detail, as proved by the case studies, the experiments, and the formal user study.
- EventRiver provides a multi-resolution visual exploration pipeline and a rich set of interactions. They allow users to explore the rich information carried by large TCTFs in a manageable way. In addition, the visualization is capable of concisely presenting the most important information in a large TCTF without clutter by modeling the data from perspective of evolution of events.
- EventRiver uses a novel incremental streaming text clustering algorithm to automatically extract TLCs, which can be mapped to real life events, from TCTFs. It also automatically characterizes the semantics and strengths over time of the TLCs and constructs TLC groups in which long-running stories and event evolution can be observed. These algorithms are standalone algorithms that can be used in other TCTF visual exploration systems.

In the future, we will extend EventRiver to support fast evolving text streams. We will also apply EventRiver to other text stream analysis applications, such as emails, weblogs, and more general online media exploration. Analyses in EventRiver will be strengthened to support visual sense making better by integrating components such as opinion analysis. In addition, we plan to support comparative analysis of multiple TCTFs in EventRiver.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Homeland security under Grand Award Number 2008-ST-108-000002. We also thank Dr.Hangzai Luo and Dr.Jianping Fan for their help on data collection and preprocessing.

REFERENCES

[1] J. Kleinberg, *Temporal Dynamics of On-Line Information Streams*. Springer, 2006.

[2] J. Allan, Ed., *Topic Detection and Tracking, Event-based Information Organization*. Kluwer Academic Publishers, 2002.

[3] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study: Final report," in *Proceedings of Broadcast News Transcription and Understanding Workshop*, 1998, pp. 194–218.

[4] Y. Yang, T. Pierce, and J. Carbonell, "A study on retrospective and on-line event detection," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 28–36.

[5] S. Havre, E. Hetzler, P. Whitney, and L. Nowell, "Themeriver: Visualizing thematic changes in large document collections," *IEEE TVCG*, vol. 8, no. 1, pp. 9–20, 2002.

[6] J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow, "Visualizing the non-visual: spatial analysis and interaction with information from text documents," in *Proc. IEEE INFOVIS*, 1995, pp. 51–58.

[7] L. Nowell, E. Hetzler, and T. Tanasse, "Change blindness in information visualization: A case study," in *Proc. IEEE INFOVIS*, 2001, pp. 15–22.

[8] R. Amar. and J. Stasko, "Knowledge task-based framework for design and evaluation of information visualizations," *Proc. IEEE Symposium on Information Visualization*, pp. 143–149, 2004.

[9] Q. Mei, "Discovering evolutionary theme patterns from text: an exploration of temporal text mining," in *KDD*, 2005, pp. 198–207.

[10] G. Fung, J. Yu, H. Liu, and P. Yu, "Time-dependent event hierarchy construction," in *KDD*, 2007, pp. 300–309.

[11] E. Hetzler, V. Crow, D. Payne, and A. Turner, "Turning the bucket of text into a pipe," in *Proc. IEEE INFOVIS*, 2005, p. 12.

[12] M. Ghoniem, D. Luo, J. Yang, and W. Ribarsky, "Newslab: Exploratory broadcast news video analysis," *Proc. IEEE VAST*, pp. 123–130, 2007.

[13] D. Fisher, A. Hoff, G. Robertson, and M. Hurst, "Narratives: A visualization to track narrative events as they develop," in *Proc. IEEE VAST*, 2008, pp. 115–122.

[14] C. Erten, P. Harding, S. Kobourov, K. Wampler, and G. Yee, "Exploring the computing literature using temporal graph visualization," in *Conference on Visualization and Data Analysis*, 2004, pp. 45–56.

[15] C. Albrecht-Buehler, B. Watson, and D. A. Shamma, "Visualizing live text streams using motion and temporal pooling," *IEEE Comput. Graph. Appl.*, vol. 25, no. 3, pp. 52–59, 2005.

[16] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins, "Visualizing tags over time," in *WWW '06: Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 193–202.

[17] J. Donath, K. Karahalios, and F. B. Viégas, "Visualizing conversation," in *HICSS '99: Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences-Volume 2*, 1999, p. 2023.

[18] S. Rose, S. Butner, W. Cowley, M. Gregory, and H. Walker, "Describing story evolution from dynamic information streams," in *VAST 2009: Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, 2009, pp. 99–106.

[19] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 497–506.

[20] H. Luo, J. Fan, J. Yang, W. Ribarsky, and S. Satoh, "Mining large-scale news video databases for supporting knowledge visualization and intuitive retrieval," *Proc. IEEE Symposium on Visual Analytics Science and Technology*, pp. 107–114, 2007.

[21] C. Aggarwal, J. Han, J. Wang, and P. Yu, "A framework for clustering evolving data streams," *VLDB*, pp. 81–92, 2003.

[22] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," in *ICDE*, 1999, pp. 512–521.

[23] WikiPedia, http://en.wikipedia.org/wiki/Main_Page, http://en.wikipedia.org/wiki/Timeline_of_Military_Operations_in_the_2006_Lebanon_War, http://en.wikipedia.org/wiki/2006_Lebanon_War, http://en.wikipedia.org/wiki/John_Mark_Karr, http://en.wikipedia.org/wiki/Timeline_of_the_2006_transatlantic_aircraft_plot, http://en.wikipedia.org/wiki/Floyd_Landis, http://en.wikipedia.org/wiki/Democratic_Party_primary,_Connecticut_United_States_Senate_election,_2006, http://en.wikipedia.org/wiki/2006_Cuban_transfer_of_presidential_duties, http://en.wikipedia.org/wiki/2006_Fox_journalists_kidnapping, http://en.wikipedia.org/wiki/Nuclear_program_of_Iran.

[24] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: Addison Wesley, 1999.

APPENDIX
MORE DETAILS ABOUT WIKIPEDIA EXPERIMENT

Here we list more example matching events from the Wikipedia experiment (Section 6.3). For each event, we cite the event description in Wikipedia and highlight the keywords appearing in the labels of the matching TLC in EventRiver in bold.

Jonbenet Murder Case

There are 11 events of this story listed in the Wikipedia timeline [23] in the time period. 20 TLCs of this story were detected from EventRiver. There were 8 matching events. We present several example matching events : (1) Wed Aug. 16, 2006, a suspect was arrested and in **custody** in **Bangkok**, **Thailand**, Wednesday for the December 26, 1996, murder of **JonBenet Ramsey**; (2) *Aug. 17 through 19, 2006*, retrospective reports on this case and the suspect **John Mark Karr** deported to U.S. from **Thailand**; (3) **Karr** was legally arrested on *Aug. 20, 2006*, after the airliner touched down at **Los Angeles International Airport**; (4) *Aug. 22, 2006*, A warrant issued by authorities in **Boulder**, Colorado, called for Karr's arrest on suspicion of first-degree murder; (5) On *Aug. 24, 2006*, "Police in California eyed **Karr** in 2001 - **Sheriff**: Suspect had 'fascination' with **JonBenet**, **Polly Klaas**".

2006 Transatlantic Aircraft Plot

9 events of this story are reported in the Wikipedia timeline [23] in the time period. 10 TLCs of this story were detected from EventRiver, as shown in Figure 2(a). There were 7 matching events. Here are several example matching events: (1) Aug. 10, 2006, "the London **Heathrow** airport detected a supposed al-Qaeda terrorist plot in which **Pakistani** suspects tried to **carry** pretended

liquid **bombings** onto the flight to America”; (2) Aug. 12, 2006, “U.S. and British sources said one of the men in **custody** in **Pakistan**, Rashid Rauf, had a key operational role in the alleged plot.”;(3) Aug. 14, 2006, Authorities in Britain and the United States imposed tight security on trans-Atlantic flights, sharply restricting what passengers could **carry** onto aircraft, where passengers are banned from bringing most liquids, **gels** or **pastes** aboard aircraft; (4) Aug. 16, 2006, Laden’s past activities in Pakistan were retrospectively reported; the link between Laden and this event was being speculated.

Others Events

Here we briefly introduce the other five major TLC groups that were compared with the Wikipedia Timeline [23] using example matching events:

- “**Cuban** transfer of presidential duties” in early Aug. 2006;
- “**Floyd Landis** is stripped of the 2006 **De France** title because of the drug use”;
- “**Joe Lieberman**’s lost to **Ned Lemond** in Democratic Party primary, Connecticut United States Senate election, 2006” in middle Aug. 2006;
- “Fox journalists **Olaf Wiig** and **Steve Centanni** were kidnapped in the **Gaza** Strip, appeared in a video tape while in capture, and released later” which happened in the end of Aug. 2006;
- “U.N. Security council demands **Iran** suspend **Uranium enrichment** by the end of Aug., 2006”.