RC25202 (W1108-030) August 8, 2011
Computer Science

IBM Research Report

ETree: Effective and Efficient Event Modeling for Real-Time Online Social Media Networks

Hansu Gu[1], Xing Xie[2], Qin Lv[1], Yaoping Ruan[3], Li Shang[1]

[1]University of Colorado at Boulder

[2]Fudan University

[3]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

ETree: Effective and Efficient Event Modeling for Real-Time Online Social Media Networks

Hansu Gu[†], Xing Xie[§], Qin Lv[†], Yaoping Ruan[#], Li Shang[†]

[†] University of Colorado at Boulder, [§] Fudan University, [#] IBM T.J. Watson Research Center

***Abstract*—Outline social media networks (OSMNs) such as Twitter provide great opportunities for public engagement and event information dissemination. Event-related discussions occur in real time and at the worldwide scale. However, these discussions are in the form of short, unstructured messages and dynamically woven into daily chats and status updates. Compared with traditional news articles, the rich and diverse user-generated content raises unique new challenges for tracking and analyzing events. Effective and efficient event modeling is thus essential for real-time information-intensive OSMNs.**

In this work, we propose *ETree*, an effective and efficient event modeling solution for social media network sites. Targeting the unique challenges of this problem, ETree consists of three key components: (1) an n-gram based content analysis technique for identifying core information blocks from a large number of short messages; (2) an incremental and hierarchical modeling technique for identifying and constructing event theme structures at different granularities; and (3) an enhanced temporal analysis technique for identifying inherent causalities between information blocks. Detailed evaluation results using 3.5 million tweets over a 5-month period demonstrate that ETree can efficiently and incrementally generate high-quality event structures and identify inherent causal relationships with high accuracy.

I. Introduction

With the fast growth of online population and rapid development of Web 2.0 technologies, online social media networks (OSMNs), which leverage both media and social networking by supporting easy web publishing and social interactions of online users, have become increasingly popular. A large amount of social media content is being generated by individual users on a daily basis. For instance, users of Twitter [1], [2], a popular microblogging social media site, send 140 million tweets per day. Moreover, OSMNs provide great opportunities for users to participate anytime and anywhere. Such user-based, real-time content generation is usually event driven. As events happen and evolve over time, users stay informed by seeking and sharing information through their social contacts (e.g., "following" and "follower" networks in Twitter). As a result, OSMNs have become the online gathering place for public engagement when real-time events happen and offer unique new opportunities for tracking and analyzing events. This has been demonstrated in various application domains, such as disease surveillance [3] and hazardous situations [4]. By sharing and receiving information among trusted and/or close social contacts, information related to specific events can be generated and disseminated in a highly effective and efficient fashion.

However, such user-generated event-related information in OSMNs is usually unstructured, and it is very difficult for individual users to capture a complete yet concise structural view of events using their social network-based information propagation channel. Moreover, as ongoing events evolve quickly and new messages are generated, the structural view of events should be adjusted to reflect the new developments in a real-time fashion. For instance, at Twitter, over 1 million tweets were generated by over 460,000 users in 128 days about the movie Avatar and every second, there may be some new updates about the event. As a result, users are constantly swamped by long streams of unstructured, redundant, and sometimes irrelevant messages, while at the same time lacking a comprehensive and well-organized view of events. *Event modeling*, which aims to identify inherent, evolving event structures and potential causal relationships, has become increasingly important for OSMNs and has the potential to significantly enhance our capabilities for information and knowledge management.

Event modeling for OSMNs is a challenging problem due to the following reasons:

- First, messages posted by users at social media sites tend to be short. For example, each tweet message has a maximal length of 140 characters. Also, messages generated by individual users tend to be unstructured, informal and differ in writing style. Such data sparseness, lack of context, and diversity of vocabulary make it difficult for traditional text analysis techniques to capture the semantic similarity among different messages [5].
- Second, different events may enjoy different popularity among users, and can differ significantly in content, number of messages and participants, time periods, inherent structure, and causal relationships [6].
- Third, large amounts of event-related information are continuously generated by OSMN users in real time. The event modeling process needs to be highly efficient, and incremental such that new information can be quickly incorporated into the event structure model.

To address these challenges, we have developed *ETree*, an effective and efficient event modeling solution targeting event-related information generated in OSMNs. Given all messages related to a specific event [1], ETree identifies the

[1]Messages related to specific events can be identified via keyword-based text search and user-based social network search. In this work, we assume messages related to an event are already identified and focus on event modeling.

major *themes* (different aspects) of the event, the key message clusters (*information blocks*) and their hierarchical structure within each theme, as well as possible causal relationships (i.e., one led to the other) between information blocks. For example, people who are interested in the Haitian earthquake event may want to track various aspects of the event, such as new statistics, rescue efforts, donation information, etc. Our solution provides updated snapshot of the event in an easy-to-read hierarchical tree structure, along with identified causalities within the event structure. Specifically, our work makes the following key contributions:

1) An n-gram based content analysis technique for identifying core information blocks from a large number of short messages;
2) An incremental and hierarchical modeling technique to efficiently identify and construct event theme structures at different granularities, which can be dynamically adjusted as events evolve;
3) An improved event life cycle analysis technique for identifying potential causalities between information blocks; and
4) A detailed evaluation study using 3.5 million tweets over a 5-month period, which demonstrates the effectiveness and efficiency of the proposed solution.

The remainder of the paper is organized as follows. We firstly survey related work in Section II. Section III presents the problem formulation and an overview of the proposed solution. Sections IV, V, and VI describe in detail the proposed techniques for information block identification, incremental construction of hierarchical theme structure, and causal relationship detection, respectively. Detailed evaluation results are presented in Section VII. Finally, Section VIII concludes the paper.

II. Related Work

This work aims to identify inherent event theme structures and causal relationships in real-time information-intensive offline social media networks. It draws upon research in several related fields, including text summarization, time-based event evolution, as well as recent research that is specific to the Twitter social media network site.

Text summarization. Much research has been conducted in the area of text summarization, focusing mostly on news data and email data. Summaries of news articles included temporal single-sentence summaries [7], centroid-based summaries of multiple documents [8], reference relationships among distinctive phrases [9] and semantic relation based key phrase extraction [10]. Fung et al. used traditional bisecting k-means clustering algorithm to model news hierarchy [11]. Trieschnigg et al. made this type of clustering algorithm more scalable for large data sets by randomly sampling the corpus [12]. The obtained event hierarchy may not be meaningful since news articles are always partitioned into a fixed number of clusters (e.g., 2). Targeting email data, Carenini et al. investigated the problem of discovering important hidden emails using fragment quotation graph and generating email summaries using clue words [13]. Our solution differs from these works in that it handles short messages (tweets), generates both summaries and hierarchical theme structures without the need to specify the number of themes beforehand, and adjusts the event models incrementally as new content is continuously generated in a real-time fashion.

Time-based event evolution. This line of work focuses on the temporal changes/relationships of events. Kleinberg identified (emerging/changing) themes in document streams (e.g., emails, research papers) based on their temporal burstiness and hierarchical structure [14]. To understand how an event emerges, changes, and disappears, Subasic et al. separated each event into several stages with equal time period and represented each stage by building a network of salient terms based on their co-occurrence frequency and time relevance [15]. From a stream of news articles, some researchers tried to detect and track new events in real time [16], [17]. The problem of identifying event causal relationships has also been investigated by researchers [6], [18]. These techniques consider both content similarity and temporal proximity in order to identify possible causal relationships in events. Our work follows this rationale, but considers the more precise temporal distribution information rather than the beginning and ending time of events. As a result, our solution achieves higher precision and recall in causal relationship identification.

Twitter-specific research. Twitter has attracted much attention in the research community during the recent years. Starbird et al. analyzed the rapid generation of Twitter communications in the Red River flood event and identified generative, synthetic, derivative and innovative properties [4]. Sakaki et al. utilized tweets as social sensors to successfully detect events like earthquake or typhoon [19]. By analyzing the top trending topics, Kwak et al. found the fast information diffusion property [20]. User intentions of using Twitter's microblogging and community services have also been studied [21], [2]. A recommendation system has been built based on both content and collaborative filtering techniques [22]. Users' tweet history is used for determining their locations, enabling better personalized services [23]. These works are complementary to our work, as they did not consider the problem of event modeling at Twitter, but nevertheless provided useful insights into the various properties of Twitter.

III. Problem Formulation and System Overview

The problem of event modeling for OSMNs can be decomposed into several sub-tasks. Given a specific event, we first collect event-related information/messages via keyword-based search (more details of this process is described in Section VII). How to detect events is beyond the scope of this paper. With scattered messages related to a certain event, we firstly cluster messages into ***information blocks*** (with high efficiency) to gain a basic understanding of the various "pieces" of an event (i.e., fundamental information units of semantically-similar messages). Next, to capture the overall structure of an event, we construct ***hierarchical theme structures***, which represent the various aspects of

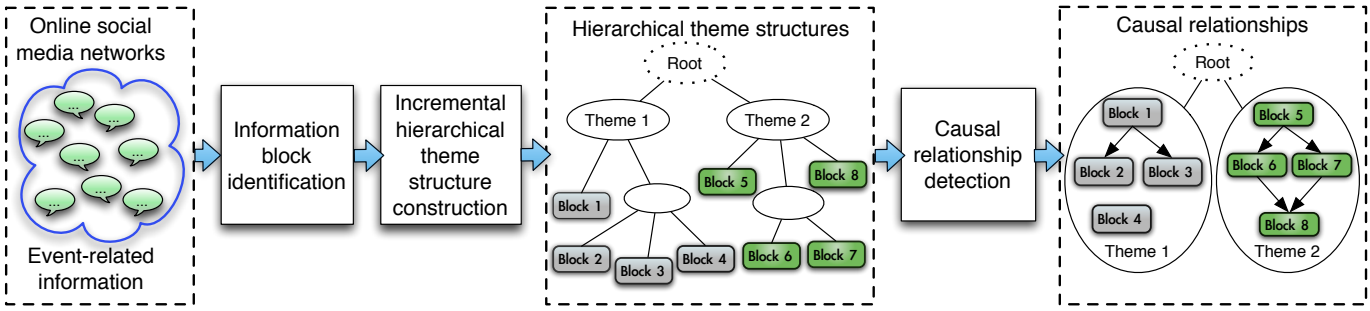

Fig. 1. ETRee: Effective and efficient event modeling for real-time online social media networks — System overview.

an event at different levels of details. Using the identified information blocks as leaf nodes, we incrementally construct hierarchical theme structures for the event. Finally, we detect potential ***causal relationship*** between pairs of information blocks. For instance, rain drenched quake survivors in the tent camps may lead to people appealing to help those slum refugees. Identifying such causal relationships within an event structure helps us to better understand how an event evolves over time. Figure 1 illustrates the workflow and key tasks of event modeling for OSMNs. A formal definition of the event modeling problem is given as follows:

Event Modeling Problem: Let E be an event and

$$E = \{(m_1, t_1), \cdots, (m_i, t_i), (m_{i+1}, t_{i+1}), \cdots\},$$

where (m_i, t_i) refers to a message m_i posted at time t_i, and messages are sorted in ascending temporal order (i.e., $t_i < t_{i+1}$). Our goal is to organize these messages into an augmented, hierarchical event tree structure, i.e., $E \Rightarrow X = \{B, H, C\}$ that consists of the following:

- **Information blocks** B**:** Each event contains a number of information blocks $B = \{b_1, b_2, \cdots\}$ and each information block $b \in B$ contains multiple messages that are semantically coherent, i.e., representing a specific semantic meaning.
- **Hierarchical theme structures** H**:** $\{B_1, B_2, \cdots\}$ are combined hierarchically and at the highest level, an event can be represented by a set of theme structures $H = \{h_1, h_2, \cdots\}$, where $h_1, h_2, \cdots$ have zero or minimum similarity. Each theme structure $h \in H$ is a hierarchical subtree of X with a few information blocks as the leaf nodes. The whole hierarchical theme structures will be adjusted as the event evolves.
- **Causal relationships** C**:** For two information blocks $b_i, b_j \in B$, if b_j is caused by b_i, then $(b_i, b_j) \in C$.

IV. Information Block Identification

Given a stream of messages that are related to a specific event, our first step is to group these messages into information blocks such that each block contains messages that share (almost) the same semantic meaning. Combining similar messages efficiently, as the goal of this step, will reduce the number of information units which will be used in the next two steps and is essential for real-time event modeling. Note that users of OSMNs can generate a huge number of messages (e.g., 140 million tweets per day at Twitter). Moreover, these messages are usually short, unstructured, and represent different writing styles of individual users. Simply clustering messages based on their cosine similarity is infeasible due to its low efficiency in large-scale message processing. To address this issue, we propose a technique that considers *key phrases* in event-specific messages.

As messages are propagated in OSMNs and new messages are generated, people tend to reuse the key phrases about an event. This is similar to the traditional "word of mouth" model, in which information is passed from one person to another. By identifying such key phrases in event-specific messages, we can then identify the core information blocks of an event.

Specifically, we propose an n-gram based content analysis technique, which works in two stages. The first stage detects frequent word sequences (i.e., n-grams, or key phrases) among a large number of event-related messages; each frequent sequence represents an initial information block. In the second stage, messages that are semantically coherent are merged into the corresponding information blocks.

N-gram based techniques have been studied previously and are considered effective and efficient for identifying word patterns in documents [24]. Given the short length of the messages people generate at OSMNs, it is important that we choose the appropriate n, which is the minimum word sequence length. Similar to the work by Leskovec et al. on identifying key phrases from news articles [9], we choose $n = 4$ as it performs well in our experiments.

Once we have identified the frequent n-grams and their corresponding information blocks, we consider the remaining messages, i.e., messages that do not contain any of the frequent n-gram patterns. For each of these messages m_i, to measure the similarity between message m_i and an information block b_j, we consider the words that belong to both m_i and b_j and calculate their TF-IDF [25] weights in b_j. We then compute the weighted cosine similarity between each message and each information block. Messages that have high similarity with some information blocks are merged into the corresponding information block. In addition, messages that belong to a specific "conversation thread" (e.g., tweets "in reply to" other tweets) are merged into the corresponding information block.

V. Incremental Hierarchical Theme Structure Construction

In this section, we present our design for constructing hierarchical theme structures using the information blocks we have identified. We describe first the static construction process, then the incremental process, which integrates newly-generated messages and information blocks into the hierarchical theme structure to keep the event structure up to date.

As we have mentioned, each event may consist of multiple themes representing the different aspects of the event, such as "rescue" and "donation" of the Haitian earthquake event; "cast", "animation" and "reviews" of the movie Avatar. Generally it is difficult to discover themes agreed by everyone, since themes could be defined differently by different people. So again a simple clustering approach does not meet the needs. Instead, each theme can be represented as a tree structure with information blocks as the leaf nodes and subtopics as the internal nodes. Such hierarchical theme structures enable a systematic organization of event-related information that is comprehensive yet concise, and allow users to explore an event from different aspects and at different granularities.

Algorithm 1 $HierarchicalStructure(B)$

Input: set of information blocks B
Output: hierarchical theme structures H
$H = \phi$
for each block $b_j \in B$ **do**
  create node $n_j = \langle b_j, M_j \rangle$; add n_j to H
end for
$< n_i, n_j, s_{i,j} >= maxSim(H)$
while similarity $s_{i,j} > 0$ **do**
  create a new parent node n_p for n_i and n_j
  $ReStructure(n_p)$
  add n_p to H
  $< n_i, n_j, s_{i,j} >= maxSim(H)$
end while
add a virtual $root$ node; return H

A. Static Theme Structure Construction

In a hierarchical theme structure, child nodes contain more specific information while parent nodes are more general and may represent the common topic of its child nodes. For instance, a parent node about "donation for earthquake" may have child nodes talking about "U.S. donation for earthquake" and "China donation for earthquake". Intuitively, the desired hierarchical theme structures should satisfy the following properties:

1) A parent node's meaning should be more general than that of its child nodes and the difference should be significant enough;
2) Nodes with similar meanings should be sibling nodes;
3) Meanings of sibling nodes should not be the same or the subset of one another.

Algorithm 2 $ReStructure(n_p)$

Input: a new parent node n_p
for each internal child node n_i of n_p **do**
  if M_i == M_p **then**
    remove n_i and attach all its children to n_p
  end if
end for
for each child pair $\langle n_i, n_j \rangle$ of n_p **do**
  if $M_i \supset M_j$ **then**
    attach node n_i to n_j as its child
  else if $M_i == M_j$ **then**
    if n_i and n_j are both internal nodes **then**
      attach n_i's children to n_j; remove n_i
    else if n_i is leaf node $\wedge$ n_j is internal node **then**
      attach n_i to n_j as its child
    end if
  end if
  ReStructure(n_j) if n_j has new child
end for

We formally define the ***Meaning*** of each individual node in the event hierarchical structure as follows:

Definition 1: A leaf node's $Meaning$ M_i is the set of keywords K_i of its corresponding information block b_i; and an internal node's $Meaning$ M_i is the intersection of its child nodes' meanings $\bigcap M_j$.

The set of keywords for each information block can be obtained by selecting the words with high TF-IDF weights. Two nodes are considered different if their $Meaning$ contain different sets of keywords.

Algorithm 1 shows the process of constructing the hierarchical theme structures. Starting with the information blocks as the leaf nodes, this process iteratively selects two nodes ($i \neq j$ and nodes n_i, n_j have no parent node) with the highest similarity using the $maxSim(H)$ procedure, and merges the two nodes into a new parent node, thus ensuring Property 2. This new parent node is then restructured (Algorithm 2) to ensure Property 1 and 3. First, if an internal child node has the same meaning as the parent node, that child node is removed and its children are attached to the parent node. Next, we examine sibling nodes. If one node's meaning is more specific than that of its sibling node, that node becomes the child of its sibling. If two sibling nodes have exactly the same meaning, then the two sibling nodes and their children are merged into one. This restructuring process continues until all nodes in the hierarchical theme structures satisfy all three properties discussed above.

The most time-consuming operation of this algorithm is the recursive restructuring process. Nevertheless, it is a local update process (only the subtree of n_p need to be restructured), which makes the algorithm more efficient than techniques that require global updates, where the whole tree need to be restructured every time a new node is added [26].

B. Incremental Structure Construction

As events happen and evolve over time, a large amount of event-related information is continuously generated by the users of OSMNs. To keep the event models up to date, newly-generated messages need to be integrated into the models in a timely fashion. It would be time-consuming and extremely wasteful if we have to reconstruct the whole structure from scratch each time a new message is added. To address this issue, we propose an incremental modeling process to maintain dynamic hierarchical theme structures.

Newly-generated messages about an event either focus on some existing topics, or contain new topics about the event. In the former case, these new messages can be easily merged into existing information blocks. While in the latter case, new information blocks need to be created; we then need to determine where to place the new blocks and adjust the overall hierarchical theme structures as needed.

To handle these changes, our incremental structure construction algorithm is designed to utilize a top-down update process: Given a new message m, we first check n_p (initially, n_p is the $root$ node), and its child nodes to select the one that is most similar to m. Note that the $Similarity()$ function is the same as the weighted cosine similarity we define in Section IV. If the most similar child node is an internal node, this process continues recursively until the node most similar to m is either n_p or a leaf node. If the similarity value is higher than a threshold δ, message m is merged into that node. Otherwise, a new node is created that contains only the new message, and the new node is added as a child (or grandchild) of n_p. After m is inserted, the $ReStructure()$ procedure is called on n_p to restructure its subtree and ensure that the three properties are still maintained.

VI. Causal Relationship Detection

Given the information blocks we have identified and the hierarchical theme structures we have constructed, one more question we want to answer in the event modeling process is whether there exists any causal relationships between information blocks. Understanding such causal relationships is important as it provides insights into how an event evolves through multiple stages and how these stages impact each other. However, finding exact causal relationships is a very difficult task without incorporating domain knowledge [6]. Previous research [6], [18] has shown that two pieces of information are more likely to be causally related if they are similar in content. Also, the study by Yang et al. [27] shows that the relevancy of two pieces of information increases when they are temporally closer to each other. Based on these observations, we aim to tackle our problem of causality detection in OSMNs by considering *both content similarity and temporal relevance*.

Given two information blocks b_i and b_j, let S and T be the content similarity and temporal relevance between these two blocks, respectively, we define the causal relationship C as a function of S and T, i.e.,

$$C = S \times T \tag{1}$$

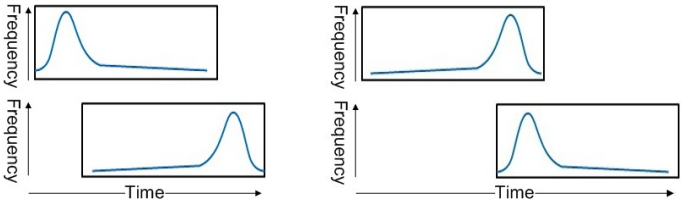

Fig. 2. Two cases demonstrating when ETree outperforms TSCAN.

For content similarity S, we use the same weighted cosine similarity as defined in Section IV. Next, we focus on defining the temporal relevance T.

Using the timestamped messages belonging to each information block, we can calculate the following temporal information of the block: (1) *start/end time*, which correspond to the timestamps of the earliest and latest messages in the block; and (2) *temporal life cycle*, which is a temporal distribution reflecting the number (or percentage) of messages posted within each time period. As the examples shown in Figure 2, by examining the temporal intersection of two information blocks, we can determine their temporal relevance in 2 stages. Considering two blocks, one on the left and one on the right of the time line, are approaching towards each other from far away. Let *critical point* be the point when two blocks overlap with each other and have the maximal temporal relevance, then

- Stage 1: Before the intersection reaches the critical point, the temporal relevance should gradually increase from minimum to maximum. In addition, the increasing speed should be different when there is no overlap (called stage $1(a)$) and when the intersection is smaller than the critical point but greater than 0 (called stage $1(b)$);
- Stage 2: After passing the critical point, the temporal relevance should gradually decrease. And when they become completely parallel (i.e., happening at the same time), the temporal relevance (and the causal relationship) decreases to minimum.

Then the question is how to determine the changing speed in different stages. Based on the analysis by Chen et al. [18], we assume that stage $1(b)$ and stage 2 have the same linear changing speed as to the intersection but the former is positive and the latter is negative. And the changing speed in stage $1(a)$ follows the inverse proportion curve.

Based on this intuition, we define the *temporal relevance* T between block b_i and b_j in Equation 2, where b_i^s and b_i^e are the start and end time of block b_i, f_t is defined as the intersection frequency of two blocks at time point t, and the range of the temporal relevance is (0,1]. Parameter $2 * \theta$ $(0 < \theta < 1)$ defines the value of the critical point. In our experiments, θ is set to 0.2 based on the power law property analysis of message generation frequency in the life cycle of information blocks.

$$T = \begin{cases} 1/\left(b_j^s - b_i^e + \frac{\theta-1}{2\theta-1}\right), & b_j^s > b_i^e \\ \frac{1}{\theta-1} * (\frac{F}{2} - 1), & 2\theta \le F < 2 \\ \frac{1}{\theta-1} * (2\theta - 1 - \frac{F}{2}), & 0 < F < 2\theta \end{cases} \tag{2}$$

where intersection

$$F = \sum_{t=b_j^s}^{min(b_i^e, b_j^e)} f_t \quad (3)$$

One work that is closely related to our causality relationship detection design is the $TSCAN$ method proposed by Chen et al. [18]. Here, we highlight the difference between our method and theirs. Firstly, the causal relationships defined in $TSCAN$ are based on blocks in two different themes and blocks within one theme are connected as a causal time line. Usually, blocks in different themes can be very different in content and most of the causal relationships identified across themes may not be useful. While within the same theme, potential actual causality between concurrent blocks is ignored by $TSCAN$. Secondly, when defining the time relevance, $TSCAN$ only utilizes the start and end time information of blocks and directly defines two blocks as having the strongest temporal relevance when there is no overlap. In our design, we leverage the detailed temporal life cycle information of blocks. And based on the observation that two blocks may have the strongest temporal relation when they have certain overlap, we utilize a parameter θ to define the overlapping critical point as described in Equation 2. By considering information blocks within the same theme structure and more accurate event life cycle distribution information, our design can achieve better characterization of temporal relevance and causal relationships than $TSCAN$, as demonstrated in the two cases in Figure 2.

In the left part of Figure 2, two blocks appear in chronological sequence which tells us these two blocks probably have strong temporal relevance. However, according to the definition in $TSCAN$, these two blocks are very close to each other when only considering their start and end time, so they are thought to be parallel and their temporal relevance is considered weak. But our design can correctly capture the strong temporal relevance by considering the small intersection of the two blocks' life cycles. In the right part of Figure 2, according to the definition in $TSCAN$, the temporal relevance is strong because the bottom block seems to appear directly after the top one. Actually their climax almost overlap, which means they probably happened in parallel, and therefore unlikely to contain causal relationship.

VII. Experimental Evaluation

In this section, we evaluate ETree, the proposed event modeling solution for OSMNs, using real-world events and event-related messages generated by individual users. Our evaluation aims to answer the following questions:

- Does our n-gram based information block identification algorithm generate coherent information blocks with good content coverage of each event?
- Does the hierarchical theme structure capture the various aspects of an event at the appropriate granularities?
- Does the incremental modeling process achieve high efficiency and generate good-quality intermediate results?
- Does our causal relationship detection algorithm achieve high accuracy with regard to the identified causalities?

A. Dataset Description

The data used in our experiments are real-world messages gathered from Twitter, one of the most popular online social media networks. Using Twitter's APIs, we have collected event-related information over a 5-month period. To ensure diversity and scalability of the evaluation data set, we manually selected 20 events spanning 7 different categories, including World, Politics, Business, Health, Entertainment, Science/Technology and Sports. The messages related to each event are collected using the keyword-based text search API provided by Twitter. Specifically, for each event, we handpick a set of keywords and use the Twitter API to collect all tweets that match at least one of the keywords. We also collected tweets that belong to the same conversation threads as the tweets returned by the search API.

A summary of the 20 events, including the number of tweets, users and days, is listed in Table I. From this table, we can see that the 20 events cover a wide range in terms of category, popularity, scale, and life span; therefore ensuring a comprehensive evaluation of the proposed solution under different scenarios. In total, our data set consists of 3.5 million tweets, and the total size of our data set is around 75GB.

B. Quality of Information Blocks and Theme Structures

First, we evaluate whether our n-gram based information block identification algorithm can capture the main content of an event, and whether the identified hierarchical theme structures have good quality.

We use $Coverage$ as the metric to evaluate the effectiveness of the information block identification algorithm. $Coverage$ of an event is defined as the percentage of messages which are captured into one of the identified information blocks. To calculate $Coverage$ of an event, we calculate the sum of the number of messages in all the information blocks of that specific event, and divide it by the total number of messages retrieved for that event. For the 20 events used in the evaluation, the information blocks identified by our method has high $Coverage$, ranging from 71% - 92% (84.2% on average).

Besides information block $Coverage$ of an event, another measure of the quality of the event structure is the relevance of the information blocks. Intuitively each theme in an event should represent a certain aspect. To evaluate the quality of identified themes of an event, we define $Coherence$ as the percentage of coherent themes in an event and a theme is considered coherent if more than half of its information blocks are relevant. To calculate $Coherence$ of an event, we manually examine the relevance of information blocks in each theme. Based on our analysis, ETree has identified highly coherent themes for almost all the 20 events, with $Coherence$ values ranging from 63% to 82% (76.9% on average).

C. Efficiency of Incremental Event Modeling

Next, we evaluate whether the incremental modeling process of ETree achieves high efficiency and generates high-quality intermediate theme structures at the same time.

TABLE I
LIST OF EVENTS USED IN EVALUATION

ID	Event	Category	#Tweets	#Users	#Days
1	Possibility and impact of China allowing its currency rising	business	718	355	14
2	MySpace's CEO was fired for not very good achievement	business	1,749	1,022	11
3	Processed foods were recalled for containing HVP	health	2,526	1,398	26
4	Apple filed a lawsuit against HTC for patent infringement	business	4,908	2,576	18
5	Duke defeating North Carolina in a match of NCAA	sports	6,279	5,321	6
6	President Obama proposed a bank reform to save financial system	politics	8,061	4,965	76
7	Many airports used body scanning to detect bombs	world	9,088	5,863	57
8	A global outbreak of H1N1 influenza virus from 2009	health	12,138	3,832	46
9	An TV series "Modern Family" were renewed for a second season	entertainment	18,692	10,549	33
10	Will Google leave China?	business	24,024	9,571	70
11	A magnitude 8.8 earthquake stroke Chile on February 27, 2010	world	31,814	13,712	41
12	Toyota had to recall millions of vehicles globally	business	43,626	15,345	71
13	Obama health care reform was passed to become law	politics	54,074	19,134	38
14	Google launched the extended release of Google Wave	science/technology	58,511	30,505	114
15	The 82nd Academy Awards ceremony was held on March 7, 2010	entertainment	87,449	49,356	33
16	2010 NBA all star game	sports	112,318	57,174	27
17	Google announced a social networking tool Google Buzz	science/technology	297,972	117,146	103
18	Apple released a table computer called iPad	science/technology	410,297	112,300	101
19	A magnitude 7.0 earthquake stroke Haiti on January 12, 2010	world	479,982	180,076	99
20	A highly popular epic science fiction film in 2009: Avatar	entertainment	1,001,530	463,128	128

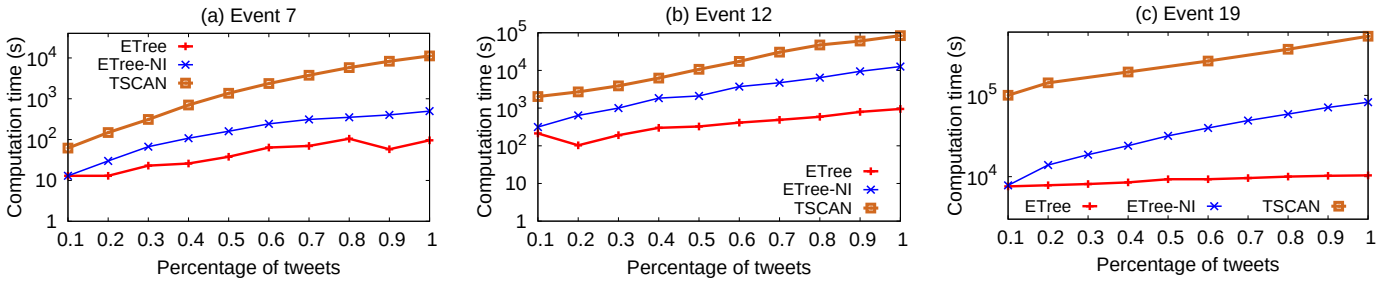

Fig. 3. Comparison of computation time of ETree, ETree-NI and TSCAN.

We compare the computation time of generating hierarchical theme structures, using three different algorithms: ETree, ETree without incremental modeling (ETree-NI), and TSCAN [18]. TSCAN is a popular algorithm widely used, which derives an event's major themes from the eigenvectors of a temporal block association matrix. Because neither ETree-NI nor TSCAN supports incremental modeling, each increment of the tweets would cause the system to re-compute the entire hierarchical structure. While in ETree, we only update the original structure by incorporating the newly-created data. Since the non-incremental algorithms take a long time to run for each event, we choose three events with different size, content and structure for this evaluation, including Event 7, 12 and 19.

Figure 3 shows the computation time in seconds for the three event modeling algorithms. The results suggest that the computation time of ETree is stable and much lower because the main influential factor is the number of newly-created tweets, while the computation time increases very quickly for ETree-NI and TSCAN. For example, for event 12, the computation time of ETree is between 100 seconds to 800 seconds for any 10% of the tweets, while the computation time of ETree-NI increases from 300 seconds to 12600 seconds and TSCAN from 2,000 seconds to 83,700 seconds as the number of tweets increases. Apparently, this dramatic increase makes it difficult for ETree-NI or TSCAN to generate up-to-date theme structure in short time intervals when the number of messages about an event becomes large. Note that we are evaluating the efficiency of the algorithms on a single core machine with limited memory, which means the absolute value of excecution time will be improved significantly within state-of-art hardware environment.

D. Quality of Detected Causal Relationships

We also evaluate how well our causal relationship detection algorithm works for different events.

Since it is difficult to manually label all actual causal relationship pairs as the ground truth, we use TSCAN and ETree to compute causal relationship pairs first, then manually verify these pairs. To reduce the influence of subjective factors in the verification process, two researchers worked independently to cross check the results. We use three metrics to quantify the results: $Precision$ measures the fraction of identified causalities that are true; $Recall$ measures the fraction of true causal relationships that are actually identified; and $F1-measure$ is defined as $2 * Precision * Recall/(Precision + Recall)$.

When we consider the quality of the identified causal relationship pairs, events with a small number of identified

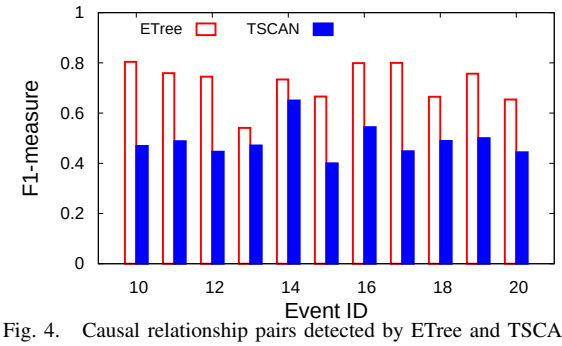

Fig. 4. Causal relationship pairs detected by ETree and TSCAN

causalities are prone to random noises (due to the limited size and information) and may easily skew the overall results. Instead, we only consider 11 popular events (event ID from 10 to 20) and report their $F1-measure$ values in Figure 4. We can see that ETree outperforms TSCAN by 49% on average for all these events.

E. Case Study

Next, we use the "Haitian earthquake" event as an example to demonstrate the quality of our theme hierarchy in more detail.

1) Hierarchical Theme Structures: We choose several easy-to-understand themes in this event and show the structures in Figure 5. Two observations can be drawn from the theme structures: (i) most information blocks in a theme have relevant content; and (ii) the hierarchical theme structure clearly reflects the level of granularity of the theme. For example, the theme chosen in the event "Haitian earthquake" talks about the rainy season after earthquake. It contains four pieces of detailed information: scientists' prediction, rain adding to misery, camps needed and only one piece plastic for every Haitian family. Each piece of the information is followed by more detailed messages. For example, block 6, 7 and 8 are more specific discussions about the content of "rain adding misery".

2) Causal Relationships: Next, we examine the causal relationship pairs that ETree has detected in this event. From the themes shown in Figure 5, we can easily find some causality pairs by reading the content. For example, block 7 talks about survivors in Haiti suffering from the rain and block 6 talks about people appealing for help. Block 7 began on Feb 12th and ended on March 14, while block 6 was from Feb 15th to March 20th. When only considering the start and end time, these blocks seem to have happened in parallel. However, when considering the life cycle distribution, we can see that the climax of block 7 was from Feb 12th to March 1st and the climax of block 6 was from March 14th to March 18th. This means when block 7's popularity began to decrease, block 6 began to become popular. These two blocks actually did not occur in parallel. This causal relationship pair clearly demonstrates the improved accuracy of ETree, compared with prior mechanisms such as TSCAN (see the first case in Figure 2, Section VI).

VIII. Conclusions

This paper presents *ETree*, an effective and efficient event modeling solution for real-time and information-intensive online social media networks. ETree utilizes an n-gram based content analysis technique to group a large number of event-related messages into semantically-coherent information blocks, an incremental modeling process to construct hierarchical theme structures, and a life cycle-based temporal analysis technique to identify potential causal relationships between information blocks. Detailed evaluation results using 20 real-world events and 3.5 millions tweets demonstrate that ETree can generate high-quality event structures with high efficiency. We anticipate to apply *ETree* to larger and more noisy dataset and identify new research problems. We are also interested in exploring more usage of social ties to improve the quality of *ETree*.

References

[1] "Twitter," http://www.twitter.com.
[2] D. Zhao and M. B. Rosson, "How and why people twitter: the role that micro-blogging plays in informal communication at work," in *GROUP '09*, 2009, pp. 243–252.
[3] C. D. Corley, D. J. Cook, A. R. Mikler, and K. P. Singh, "Text and structural data mining of influenza mentions in web and social media," *International Journal of Environmental Research and Public Health*, vol. 7, no. 2, pp. 596–615, 2010.
[4] K. Starbird, L. Palen, A. L. Hughes, and S. Vieweg, "Chatter on the red: what hazards threat reveals about the social life of microblogged information," in *CSCW '10*, 2010, pp. 241–250.
[5] D. Metzler, S. Dumais, and C. Meek, "Similarity measures for short segments of text," *Lecture Notes in Computer Science*, vol. 4425, p. 16, 2007.
[6] R. Nallapati, A. Feng, F. Peng, and J. Allan, "Event threading within news topics," in *CIKM '04*, 2004, pp. 446–453.
[7] J. Allan, R. Gupta, and V. Khandelwal, "Temporal summaries of new topics," in *SIGIR '01*, 2001, pp. 10–18.
[8] D. R. Radev, H. Jing, M. Stys, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing & Management*, vol. 40, no. 6, pp. 919 – 938, 2004.
[9] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *KDD '09*, 2009, pp. 497–506.
[10] X. Wu, G.-Q. Wu, F. Xie, Z. Zhu, and X.-G. Hu, "News filtering and summarization on the web," *IEEE Intelligent Systems*, vol. 99, no. PrePrints, 2010.
[11] G. P. C. Fung, J. X. Yu, H. Liu, and P. S. Yu, "Time-dependent event hierarchy construction," in *KDD '07*, 2007, pp. 300–309.
[12] D. Trieschnigg and W. Kraaij, "Hierarchical topic detection in large digital news archives," in *Proc. of the 5th Dutch Belgian Information Retrieval workshop*, 2005, pp. 55–62.
[13] G. Carenini, R. T. Ng, and X. Zhou, "Summarizing email conversations with clue words," in *WWW '07*, 2007, pp. 91–100.
[14] J. Kleinberg, "Bursty and hierarchical structure in streams," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373–397, 2003.
[15] I. Subasic and B. Berendt, "Web mining for understanding stories through graph visualisation," in *ICDM '08*, 15-19 2008, pp. 570 –579.
[16] T. Brants, F. Chen, and A. Farahat, "A system for new event detection," in *SIGIR '03*, 2003, pp. 330–337.
[17] C. Wang, M. Zhang, S. Ma, and L. Ru, "Automatic online news issue construction in web environment," in *WWW '08*, 2008, pp. 457–466.
[18] C. C. Chen and M. C. Chen, "TSCAN: a novel method for topic summarization and content anatomy," in *SIGIR '08*, 2008, pp. 579–586.
[19] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *WWW '10*, 2010, pp. 851–860.
[20] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *WWW '10*, 2010, pp. 591–600.
[21] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *WebKDD/SNA-KDD '07: Proc. of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, 2007, pp. 56–65.
[22] J. Hannon, M. Bennett, and B. Smyth, "Recommending twitter users to follow using content and collaborative filtering approaches," in *Proc. of the fourth ACM conference on Recommender systems*, ser. RecSys '10, 2010, pp. 199–206.

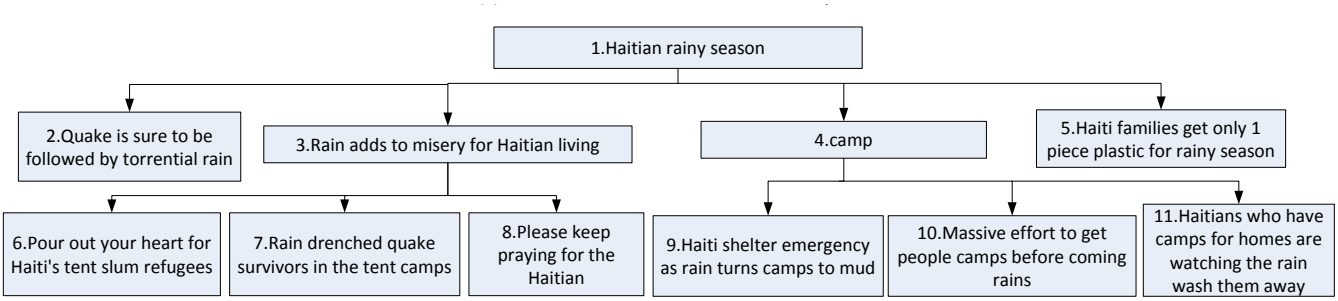

Fig. 5. Partial hierarchical theme structures constructed by ETree for the event "Haitian earthquake".

[23] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *CIKM '10*, 2010, pp. 759–768.

[24] W. Cavnar and J. M. Trenkle, "N-gram-based text categorization," in *SDAIR '94: Proc. of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161–175.

[25] G. Salton, E. A. Fox, and H. Wu, "Extended boolean information retrieval," *Commun. ACM*, vol. 26, no. 11, pp. 1022–1036, 1983.

[26] D. H. Widyantoro, T. R. Ioerger, and J. Yen, "An incremental approach to building a cluster hierarchy," in *ICDM '02*, 2002, p. 705.

[27] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *SIGIR '98*, 1998, pp. 28–36.