Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 1340973, 10 pages
http://dx.doi.org/10.1155/2016/1340973

Research Article

Mining Sequential Update Summarization with Hierarchical Text Analysis

Chunyun Zhang,[1] Zhongwei Si,[2] Zhanyu Ma,[2] Xiaoming Xi,[1] and Yilong Yin[3]

[1]*School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, Shandong 250014, China*
[2]*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*School of Computer Science and Technology, Shandong University, Jinan, Shandong 250014, China*

Correspondence should be addressed to Zhanyu Ma; mazhanyu@bupt.edu.cn

Received 1 October 2015; Revised 21 December 2015; Accepted 5 January 2016

Academic Editor: Yassine Hadjadj-Aoul

The outbreak of unexpected news events such as large human accident or natural disaster brings about a new information access problem where traditional approaches fail. Mostly, news of these events shows characteristics that are early sparse and later redundant. Hence, it is very important to get updates and provide individuals with timely and important information of these incidents during their development, especially when being applied in wireless and mobile Internet of Things (IoT). In this paper, we define the problem of sequential update summarization extraction and present a new hierarchical update mining system which can broadcast with useful, new, and timely sentence-length updates about a developing event. The new system proposes a novel method, which incorporates techniques from topic-level and sentence-level summarization. To evaluate the performance of the proposed system, we apply it to the task of sequential update summarization of temporal summarization (TS) track at Text Retrieval Conference (TREC) 2013 to compute four measurements of the update mining system: the expected gain, expected latency gain, comprehensiveness, and latency comprehensiveness. Experimental results show that our proposed method has good performance.

1. Introduction

Internet of Things (IoT) is a new type of the Internet. It is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data [1]. Many high-tech companies over the world have already started developing IoT products and services and promoting their early stage of IoT products and services in a number of market domains. Among the most notable challenges, wireless and mobile technologies are the underlying technologies for realizing the IoT [2, 3]. Resource constrained devices are required to communicate with other devices in wireless networks. The devices are also required to communicate on the move. In addition to these requirements, various technical and scientific research considerations are also required. One of the key techniques is to develop semantic and intelligent web for IoT [4]. The core of this technique is the combination of the traditional Internet technologies and the wireless and mobile technologies. For example, when an unexpected news event occurs, such as natural disaster (e.g., earthquake) or human accidents (e.g., air crash), some event data can be collected by IoT devices, and they submit these event data to the Internet. And these data in the Internet will form some real-time news. Based on effective sequential update summarization system, the IoT system can send individuals useful, new, and timely updates by mobile devices. Hence, developing effective sequential update summarization techniques is very important for the IoT.

However, due to the special characteristic of unexpected news event, it is a big challenge to construct an effective sequential update summarization system. Mostly, the information about unexpected news events is rapidly developing [5]. For instance, immediately after the outbreak of an unexpected event, the corpus may be sparsely populated with relevant news. Even when, after a few hours, relevant

news is available, it is often inaccurate or highly redundant. That is because news of the event is widely spread through multilevel news channels around the world. However, based on the diversity of journalistic sources, details reported about the event are redundant, dynamic, and sometimes mistaken. Furthermore, it becomes much harder to gather authoritative news, when facing major events which involve extensive damage to life or crippling of infrastructure. This may cause rumors and unsubstantiated information to propagate [6]. Meanwhile, the sudden events are also very important topics to individuals. People want to get timely information, especially for these people who are relative to these sudden events; they even cannot afford waiting for comprehensive reports to materialize [7].

Unfortunately, existing solutions cannot satisfy people's demands in getting useful, new, and timely sequential update summarizations about these events. That is because the problem of sequential update summarization extraction refers to techniques intercrossed among text summarization, topic detection and tracking, and time-based summarization. However, most current summarization systems can either use static summarization methods [8–13] or use topic detection and tracking (TDT) methods [14–18]. These methods only provide sentences extracted with particular properties based on traditional techniques of natural language processing (NLP) [19] or only provide topic-level summaries. In most ways, the sequential update summarization is an event- and sentence-level analogue of "first topic detection" problem [20]. In all, there is no support for only presenting people with novel content (i.e., updates to the user) and updates can suffer from poor coverage and unreliable information.

In this paper, we define the problem of sequential update summarization extraction for unexpected news events. This task can be considered as a variation of topic detection and tracking and time-based document summarization. Hence, the problem definition, the evaluation, and the used method are based on these techniques. With significant extension of the abovementioned techniques, we present a new hierarchical summarization system, which focuses on extracting sequential update summarization on unexpected news events. The system tries to broadcast with useful, new, and timely sentence-length updates about a developing event by incorporating the technologies of time-based topic-level and sentence-level summarization. With the application to the sequential update summarization (SUS) task of temporal summarization (TS) track [21] at Text Retrieval Conference (TREC) [22], we evaluated the effectiveness of our new method in view of precision, recall, timeliness, and novelty of updates. By computing the expected gain, expected latency gain, comprehensiveness, and latency comprehensiveness (evaluation metric of SUS task) of our extracted updates of 10 topics, we conclude that our proposed method has a good performance.

The contributions of this paper are threefold:

(a) A general definition of problem SUS is proposed.

(b) A novel framework for SUS that incorporates the technologies of time-based topic-level and sentence-level summarization is introduced.

(c) An application of this framework to the sequential update summarization task of temporal summarization (TS) track is implemented.

In the rest of this paper, we firstly review some related work on information retrieval and text summarization in Section 2. Then, we formalize the problem of sequential update summarization extraction in Section 3. In Section 4, we present the novel hierarchical update mining system and we introduce the evaluation criteria in Section 5. We conduct experiment to verify the effectiveness of our proposed method in Section 6 and conclude in Section 7.

2. Related Work

The problem of sequential update summarization has its roots in topic detection and tracking [23], time-based summarization techniques [14, 20], and multidocuments summarization [9, 24, 25].

2.1. Topic Detection and Tracking. Topic detection and tracking (TDT) refers to the document-level tasks which associated with detecting and tracking news events [23]. It is a body of research and an evaluation paradigm that addresses event-based organization of broadcasting news.

Authors of [20] suggested retrospectively selecting novel and relevant sentences from a stream of news articles. However, the TDT is more topic based than sentence based. In most ways, the sequential update summarization is an event- and sentence-level analogue of TDT's "first topic detection" problem [20].

Referring to the time-based summarization as the task of temporal summarization, most of these systems focus on temporal expression extraction from text normalizing references to dates, times, and elapsed times [14]. The system in [26] generated the meaningful temporal summarization of event-related updates and automatically annotates the identified events in a timeline. Methods proposed in [27] retrieved sequential versions of a single web page during predefined time intervals. The paper [28] presented a framework that extracts events relevant to a query from a collection of documents and placed these events along a timeline.

2.2. Multidocuments Summarization. Text summarization techniques leverage a wide range of information retrieval (IR) and natural language processing (NLP) techniques. Some focus primarily on techniques that have been developed in IR [25], while most try to leverage both IR approaches and some aspects of NLP [19]. As one of the subproblems of text summarization, multidocument summarization (MDS), which refers to the task of generating a text summary of a pool of documents on the same topic, includes two broad approaches: extractive summarization and abstractive summarization. The extractive summarization extracts summary which consists of sentences extracted from the pool of documents, while the abstractive summarization extracts summary generated based on the pool of documents.

The core technique of the extractive summarization research is to summarize a body of texts by extracting

sentences that have particular properties. Sentence extraction techniques consider the words in the sentences, look for cue words and phrases [11, 24], consider even more focused features such as sentence length and case of words [29], or compare patterns of relationships between sentences [30–32]. Most of these approaches use statistics from the corpus itself to decide on the importance of sentences, and some leverage existing training sets of summaries to learn the properties of a summary [29, 33]. Other methods computed sentence importance based on the eigenvector of a graph representation of sentences [34].

Methods investigated in this paper are mainly similar to extractive summarization. The goal of our proposed method is to extract time-based sentences which have high confidence.

3. Problem Definition

The problem of sequential update summarization has been investigated in many literatures. However, until now, there is still no clear definition on it. In this section, we will give a general definition on the problem of sequential update summarization as follows.

An unexpected event, e, is a temporally acute topic with a clear onset time, $[t_s, t_e]$. An event query, Q_e, is the representation of the event description expressed by a user during the event. The set of keywords associated with the event, $\mathcal{K}(e)$, represents the important information that should be included in updates to deliver to users (e.g., the location where the event happened, the death number caused by this event). The system observes a temporally ordered stream of documents, $[d_1, d_2, \ldots]$. On the observation of d_t, the system makes a decision to emit zero or more updates. The pool of candidate updates consists of sentences in documents comprised of the most recent k documents in the event timeframe. Figure 1 illustrates a schematic diagram of the sequential update summarization system. Based on the schematic diagram of Figure 1, we present a general framework of sequential update summarization in Algorithm 1. According to Algorithm 1, an effective sequential update summarization system should be supported by time-sensitive information retrieval technique, accurate keywords mining method, and effective updates scoring algorithm.

Require:
SequentialUpdateSummarization $\{S, C, Q_e, t_s, t_e\}$:
S = the SUS system;
C = time-ordered corpus;
Q_e = keyword query of a sudden event;
t_s = start time of a sudden event;
t_e = end time of a sudden event;
Ensure: updates set U
(1) $U \leftarrow \{\}$;
(2) S: $Retrieval(Q_e)$
(3) **for** $d \in C$ **do**
(4) *do*
(5) $S : Process(d)$;
(6) $t \leftarrow d.Time()$;
(7) **if** $t \in \{t_s, t_e\}$ **then**
(8) *then*
(9) $U_t \leftarrow S.Decide$;
(10) **for** $u \in U_t$ **do**
(11) *do*
(12) $U.Append(u; t)$;
(13) **end for**
(14) **end if**
(15) **end for**

Algorithm 1: Sequential Update Summarization System.

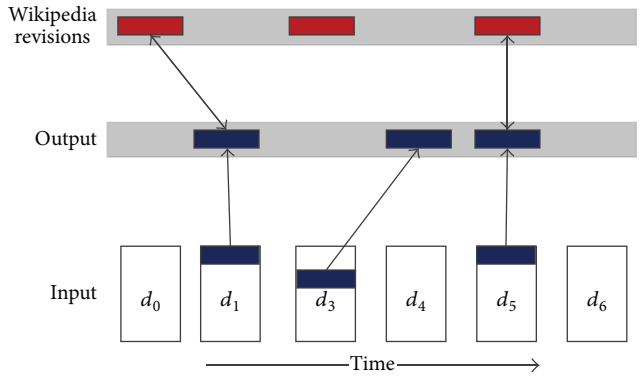

Figure 1: The sequential update summarization system observes a buffered stream of documents and makes decisions based on the contents of the input buffer and the timestamp of these documents to form the real update, which actually are time-based Wikipedia revisions.

4. Hierarchical Sequential Update Summarization System

To investigate update mining methods on unexpected events, we construct a hierarchical sequential update summarization mining system in this section. The framework of the system is illustrated in Figure 2. The framework contains three main modules: preprocessing and information retrieval module, keywords mining module, and sentence scoring module. The first module makes sure the event-relevant documents are time sensitive. The second module extracts time-based event-relevant keywords by using the hierarchical text analysis techniques. The third module focuses on scoring novel sentence-level updates.

4.1. Preprocessing and Information Retrieval Module. Because the original dataset is processed with some specific technologies, such as encryption, compression, and serialization [37], the system should firstly do some preprocessing on the available data and extract event-relevant document during each timeframe. The overall process of this module is described as follows:

(i) *Decrypt File*. The first step is to decrypt the files using the authorized key from authority. This step converts the GPG file format to SC file format.

(ii) *Deserialization*. We use stream corpus toolbox to parse these SC files to TXT files. The authority of

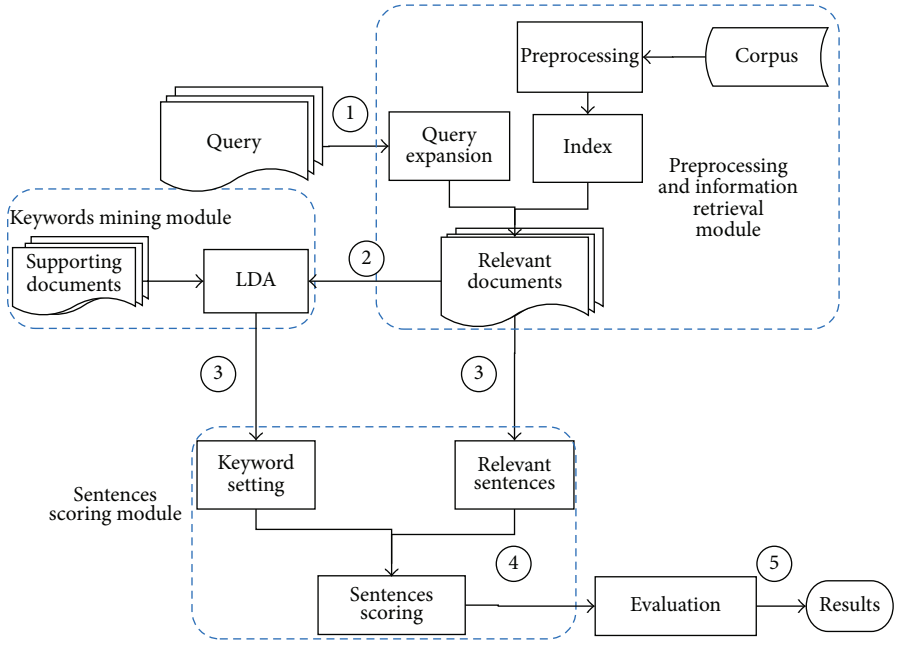

Figure 2: The framework of the hierarchical sequential update summarization system [35].

TREC provided the stream corpus toolbox to parse these SC files. The stream corpus toolbox gives a common data interchange format for document processing pipelines, which applies language processing tools to large streams of text.

(iii) *Build Index*. To obtain topic-relevant documents from large stream documents, the index of these big data should be built. This step is to build index by Indri [38] for query-based information retrieval. Indri is one of the mostly used search engines in information retrieval domain, which combines inference networks with language modeling. The query language of Indri, which is reminiscent of the Inquery query language, allows researchers to experiment with proximity, document structure, text passages, and other document features without writing code.

(iv) *Information Retrieval*. The last step is to use Indri as a tool for information retrieval. Given an event query Q_e, Indri returns all ranked relevant documents according to their responding confidence computed by the criterion of Indri. This step enables users to submit the queries and obtain the most relevant documents in each timeframe.

4.2. Keywords Mining Module. In this module, we utilize hierarchical Latent Dirichlet Allocation to find potential topics and return the most representative words of each topic as keywords.

Latent Dirichlet Allocation (LDA) [36] is a statistical model, specially a topic model, which can be used to identify hidden topic from a large document collection corpus. The basic idea of LDA is that a document can be considered as a mixture of a limited number of topics and each meaningful word in the document can be associated with one of these topics. Given a corpus of documents, LDA attempts to identify a set of topics, associate a set of words with a topic, and define a specific mixture of these topics for each document in the corpus. A thorough and complete description of the LDA model can be found in [36]. The vocabulary for describing the LDA model is as follows:

(i) *Word*. A word is a basic unit defined to be an item from a vocabulary of size W.

(ii) *Document*. A document is a sequence of n words denoted by $d = (w_1, \ldots, w_n)$, where w_n is the nth word in the sequence.

(iii) *Corpus*. A corpus is a collection of M documents denoted by $D = (d_1, \ldots, d_M)$.

In the statistical natural language processing, it is common to model each document d as a multinomial distribution θ_d over T topics and each topic z_j, $j = 1 \cdots T$, as a multinomial distribution $\phi^{(j)}$ over the set of words W. In order to discover the set of topics used and the distribution of these topics in each document in a corpus of documents D, we need to obtain an estimate of ϕ and θ. Blei et al. [36] have shown that the existing techniques of estimating ϕ and θ are slow to converge and propose a new model LDA. The LDA based model assumes a prior Dirichlet distribution on θ, thus allowing the estimation of ϕ without requiring the estimation of θ.

LDA assumes a generative process for creating a document [36] as presented below:

(i) Choose $N \sim Poisson(\xi)$: select the number of words N.

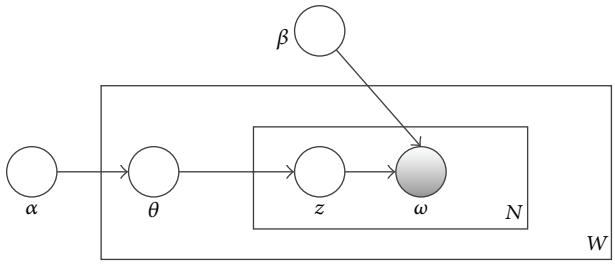

Figure 3: The probabilistic graphical model of Latent Dirichlet Allocation (LDA) [36].

(ii) $\theta \sim Dir(\alpha)$: select θ from the Dirichlet distribution parameterized by α.

(iii) For each $w_n \in w$,

(a) choose topic $z_n \sim Multinomial(\theta)$;

(b) choose a word (w_n) from $p(w_n \mid z_n, \beta)$, a multinomial probability ϕ^{z_n}.

In this model, various distributions, namely, the set of topics, topic distribution for each of the documents, and word probabilities for each of the topics, are in general intractable for exact inference [36]. The probabilistic graphical model of LDA is illustrated in Figure 3. The joint probability distribution of LDA is

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1} p(z_n \mid \theta) \cdot p(w_n \mid z_n, \beta). \quad (1)$$

Hence, a wide variety of approximate algorithms are considered for LDA. These algorithms attempt to maximize likelihood of the corpus given the model. A few algorithms have been proposed for fitting the LDA model to a text corpus such as variational Bayes [36, 39–41], expectation propagation [42], and Gibbs sampling [43].

In this paper, for each event in each hour, we firstly retrieve the most 500 relevant documents and then extract keywords by LDA in current hour. In this module, we use the GibbsLDA++ tool [44] to extract keywords. We firstly use the LDA toolkit to discover two topics and choose the most representative words for each topic; secondly, we discover 5 new topics by the same method under the topic discovered in the last step and choose the most representative words of each topic; lastly, we integrate the two level representative words of each topic to form keywords set $\mathcal{K}(e)$.

4.3. Sentences Scoring Module. We utilize three sentence scoring methods in this module: KLP method, SKD method, and KS method [45].

The first method assumes an update is a long sentence which should shoot many keywords and be placed on the first place in a paragraph. Hence, it considers three important factors: the keywords diversity, the length of a sentence, and the position of the sentence, which we named KLP method. The scoring metric is as follows:

$$\text{Score}(s_i) = \alpha \frac{\sum_{w \in s_i} tf(w) \cdot idf(w)}{\max_{s_j \in d} \left\{ \sum_{w \in s_j} tf(w) \cdot idf(w) \right\}} + \beta \frac{\text{Length}_{s_i}}{\max_{s_j} \left\{ \text{length}_{s_j} \right\}} + \gamma \, \text{position}_{s_i}, \quad (2)$$

where $w \in \mathcal{K}(e)$ is one of the keywords of event e extracted in Section 4.2 and α, β, and γ are weights of the keyword diversity, length, and position, respectively. When computing $idf(w)$, the documents are referred to relevant documents in the current hour. If a sentence is placed on the first place of a paragraph, $\text{position}_{s_i} = 1$, or $\text{position}_{s_i} = 0$.

The second method assumes that an update should be a short length sentence with larger keywords diversity, because a too long sentence is normally a retrospective summary of an event, not an update. We named this metric SKD, whose scoring metric is as follows:

$$\text{Score}(s_i) = \frac{1}{N(N+1) \cdot \text{Length}} \cdot \sum_{j=1}^{k-1} \frac{\text{Score}(w_j) \cdot \text{Score}(w_{j+1})}{\text{distance}(w_j, w_{j+1})}, \quad (3)$$

where N is the number of keywords included in s_i, $\text{Score}(w)$ is the confidence of keyword w obtained from Section 4.2, and $\text{distance}(w_j, w_{j+1})$ is the distance between w_j and w_{j+1}.

The third method is a keyword shooting method, which only considers the diversity of keywords included in the sentence. We named it KS method. Its scoring metric is as follows:

$$\text{Score}(s_i) = \frac{\left| V_{\text{keywords}} \cap s_i \right|}{s_i}, \quad (4)$$

where V_{keywords} is the keyword vector of the event e. s_i is the ith related sentences of event e.

After getting high confidence sentences, the postprocessing module will do the duplicate removal to sentences, which first finds the same sentences with different sentences ID and then compares the stream ID of all sentences and chooses the one with the earliest time information as the submission sentence.

5. Evaluation Method

Document summaries are difficult to evaluate, because all results in minor variations, such as rewording portions of the summary, reordering the sentences, and omitting dubiously important information, are still excellent summaries. The most popular summary evaluation method is comparing agreement between sentences selected by experts and sentences selected by computer [9, 46], or comparing agreement in the ranks of sentences that a system generates [47].

However, since comparing based on some sentence variant is difficult, we introduce the concept of gold nugget, which is defined as atomic novel pieces of information relevant to unexpected events. For example, in the task of SUS, gold nuggets are text perceived as relevant and novel for the edit of Wikipedia articles. Each gold nugget $n \in N$ is assigned with an importance grade by annotators: $R : N \rightarrow [0, 1]$. Hence, we can compare extracted updates with these atomic gold nuggets in a more accurate manner.

Traditional IR and text summarization evaluations are concerned with the quality and the quantity of relevant materials. In this paper, the sequential update summarization system focuses on the following properties:

(i) Updates are relevant sentences to the unexpected events.

(ii) Updates should be novel which must match with at least one gold nugget and can be matched with several gold nuggets.

(iii) Updates are sentences which are early extracted from event-relevant news. The first sentence about an event is clearly novel; the earlier the time of the first sentence of an event, the lower the latency of the update.

(iv) Updates are short sentences which should not be too verbose.

That is to say, we want to measure the relevance, latency, verbosity, and matching of the extracted updates. To measure the abovementioned properties, the SUS task of TREC defined the measurement of four parameters: expected gain, latency expected gain, comprehensiveness, and latency comprehensiveness [48].

Before introducing the definition of the four parameters, we firstly explain some fundamental definitions. Given an update set U and a gold nugget n, the matching function between them is

$$M(n, U) = \underset{\{n \in U : n \approx u\}}{\operatorname{argmin}} u \cdot t. \tag{5}$$

Besides the matching function, two discounts are defined to evaluate the timeliness and conciseness of the extracted updates set: latency discount and verbosity discount. Given a nugget whose timestamp is t', the latency discount is a latency penalty $L_d(t', t)$, which is a monotonically decreasing function of $t - t'$. Similarly, verbosity discount is also a penalty function $V_d(u)$, which is defined as a string length penalty function, monotonically decreasing in the number of words of the update string. Based on the abovementioned concepts, the discounted gain between an update u and a matching nugget n is

$$g(u, n) = R(n) * \text{discount factor}, \tag{6}$$

where the discount factor can be latency discount, verbosity discount, or the compound of the two discounts (e.g., $L_d * V_d$).

Hence, the overall expected gain is similar to traditional notions of precision in IR. It is defined as

$$\text{MEG} = \frac{1}{|E|} \sum_{e \in E} \text{EG}\left(U^e\right), \tag{7}$$

where E is the set of evaluation events and U^e is the system submission for event e, and EG(U) is defined as

$$\text{EG}(U) = \frac{1}{\sum_{u \in U} V(u)} \sum_{\{n \in N : M(n,U) \neq \phi\}} g(M(n, U), n). \tag{8}$$

To evaluate the system performance on the time after an event, the latency gain is defined as the time-sensitive expected gain for the first τ seconds as

$$\text{EG}_\tau(U) = \text{EG}\left(U_\tau\right). \tag{9}$$

In addition to good expected gain, the performance of providing a comprehensive set of updates is also very important. That is to say, the more nuggets the extracted updates set covers, the better the system performs. It is similar to traditional notions of recall in information retrieval evaluation. Given a set of system updates, the comprehensiveness is similar to the recall of IR, which evaluates the coverage on gold nuggets as

$$C(U) = \frac{1}{\sum_{n \in N} R(n)} \sum_{\{n \in N : M(n,U) = \phi\}} g(M(n, U), n). \tag{10}$$

Similarly, the latency comprehensiveness is a time-sensitive notion of comprehensiveness as follows:

$$C_\tau(U) = C\left(U_\tau\right). \tag{11}$$

6. Experimental Results and Discussions

6.1. Data and Topics. The data used in the SUS task of TS track is provided by the Organizer of KBA track [49] at TREC, which is hosted by Amazon Public Dataset service. This corpus [50] consists of a set of timestamped documents from a variety of news and social media sources covering the time period October 2011 through January 2013, whose time span is 17 months with 11,248 hours. There are more than 1 billion documents, each with absolute timestamp that places it in the stream, which is mainly composed of news, social (blog, forum), and web (e.g., arxiv, linking events) content. All documents contain a set of sentences, each with a unique identifier.

There are 10 events/topics (listed in Table 1) [51] in the SUS task; each has a single type title, description (URL to Wikipedia entry), beginning and end times, and query keywords. Types are taken from {accident, shooting, storm, earthquake, bombing} and they have a set of attributes, such as location, death, and financial impact. Algorithm 2 illustrates the definition of the event of "2012 Buenos Aires Rail Disaster." For each sudden event query, we chose the top 500 relevant documents returned by Indri as the relevant documents of each sudden event query in one hour.

6.2. Results. We applied our hierarchical update mining system on the overall ten topics. For each topic, to evaluate these extracted updates, we chose the top 60 updates as the assessment data due to their confidences computed by the KLP, SKD, and KS method. The evaluation processes were

Table 1: Queries and titles of 10 topics of temporal summarization track [51].

Query of topics	Title of topics
(1) Buenos aires train crash	2012 Buenos Aires Rail Disaster
(2) Pakistan factory fire	2012 Pakistan garment factory fires
(3) Colorado shooting	2012 Aurora shooting
(4) Sikh temple shooting	Wisconsin Sikh temple shooting
(5) Hurricane isaac	Hurricane Isaac (2012)
(6) Hurricane sandy	Hurricane Sandy
(7) Midwest derecho	June 2012 North American derecho
(8) Typhoon bopha	Typhoon Bopha
(9) Guatemala earthquake	2012 Guatemala earthquake
(10) Tel aviv bus bombing	2012 Tel Aviv bus bombing

```
<event>
  <id>1</id>
  <start>1329910380</start>
  <end>1330774380</end>
  <query>buenos aires train crash</query>
  <type>accident</type>
  <locations/>
  <deaths/>
  <injuries/>
</event>
```

Algorithm 2: An unexpected event definition for "2012 Buenos Aires Rail Disaster" in the SUS task [48].

mainly gold nuggets extraction and update-nugget matching. In this paper, the gold nugget was extracted by assessors by reading all edits of the Wikipedia article for each topic, manually extracting text perceived as relevant and novel for that edit. Additionally, they assigned an importance grade to every text fragment, or nugget, and noted any dependencies in the information. The update-nugget matching refers to matching our extracted updates to these gold nuggets to evaluate their accuracy and coverage of the information. The latency discount function and the verbosity discount function [48] used in this paper are

$$L(n \cdot t, u \cdot t) = 1 - \frac{2}{\pi} \arctan\left(\frac{u \cdot t - n \cdot t}{\alpha}\right),$$
$$V(u) = 1 + \frac{|u| - \left|\bigcup_{n \in M^{-1}(u,U)} M(n,U)\right|}{\text{avg}_{n \in N} |n|}, \quad (12)$$

where $\alpha = 3600 * 6$ is the latency step (6 hours) and $|u|$ and $|n|$ are the length (in number of words) of the update u and nugget n. By applying the abovementioned functions on evaluation metric introduced in Section 5, we computed the four performance parameters.

Table 2: The μ and σ of expected gain and expected latency gain over all events of the multi-level SUS system. (The E [gain] is the expected gain which is similar to traditional notions of precision in information retrieval; E [latency gain] is the time-sensitive expected gain).

Methods	E [Gain]	E [latency gain]
The best reported	***0.149*** (***0.101***)	***0.136*** (***0.090***)
ICTNET-run2	*0.102* (*0.045*)	*0.127* (*0.075*)
ICTNET-run1	*0.101* (*0.045*)	*0.125* (*0.075*)
Mid-value	*0.053* (*0.041*)	*0.067* (*0.057*)
KS	0.149 (0.101)	0.136 (0.090)
SKD	0.103 (0.084)	0.103 (0.050)
KLP (0.6, 0.2, 0.2)	0.071 (0.039)	0.074 (0.031)
KLP (0.5, 0.2, 0.3)	0.065 (0.034)	0.067 (0.026)
KLP (0.5, 0.3, 0.2)	0.065 (0.034)	0.067 (0.026)

Table 3: The μ and σ of comprehensiveness and latency comprehensiveness over all events of the multi-level SUS system (Comprehensiveness is similar to recall in IE, which evaluates coverage of nuggets; latency Comp. is the time-sensitive comprehensiveness).

Methods	Comprehensive	Latency Comp.
The best reported	***0.445*** (***0.191***)	***0.571*** (***0.358***)
UWaterloo-rg2	*0.441* (*0.198*)	*0.562* (*0.349*)
UWaterloo-qlec2t25	*0.433* (*0.170*)	*0.537* (*0.322*)
Mid-value	*0.204* (*0.146*)	*0.260* (*0.217*)
KLP (0.5, 0.3, 0.2)	0.224 (0.178)	0.292 (0.270)
KLP (0.5, 0.2, 0.3)	0.224 (0.178)	0.288 (0.262)
KLP (0.6, 0.2, 0.2)	0.204 (0.146)	0.260 (0.217)
SKD	0.131 (0.138)	0.176 (0.203)
KS	0.099 (0.099)	0.126 (0.164)

In addition to our previously reported results [35], Tables 2 and 3 illustrate some results reported by the SUS task of TREC 2013 and the five results of these three methods. The four parameters are evaluated by comparing generated updates with gold nuggets by using expected gain, expected latency gain, comprehensiveness, and latency comprehensiveness metrics. The expected gain is similar to traditional notions of precision in IR. Expected latency gain is a time-sensitive expected gain. Comprehensiveness is similar to recall in IR, which evaluates the coverage of gold nuggets. The latency comp. is the time-sensitive comprehensiveness [48]. The results in italics are the top 3 and the midvalue results based on corresponding parameters, which are reported in the SUS task in 2013 [48].

Table 2 illustrates the top three and the midvalue results of the expected gain and the latency expected gain in the SUS task in TREC 2013. The ICTNET-run2 and ICTNET-run1 [52] are results submitted by the Institute of Computing Technology, Chinese Academy of Sciences. They firstly chose event-relevant sentences and decided a sentence as an update if it includes words of a handpicked trigger word list, such as kill and death. From Table 2, we can see that the KS method has the best expected gain and expected latency gain, which are equal to the best reported results. That is because

the keywords list of KS method is generated by hierarchical LDA method, which can generate much more accurate keywords lists compared with the man-made keywords list of ICTNET methods. Hence, the KS method is superior to the two ICTNET methods. Table 2 also illustrates that the expected gain and latency gain of the KLP and SKD are all above the midvalue result, which shows that the KLP and SKD methods are effective methods in extracting updates of unexpected news events. By comparing the results of our three investigated methods in Table 2, we can conclude that the KS method is the most effective method in evaluating the metric of expected gain and expected latency gain; for example, it can extract updates in an accurately and timely manner.

Table 3 illustrates the top three and the midvalue results of comprehensiveness and latency comprehensiveness in the SUS task in TREC 2013. The reported top three results are submitted by the University of Waterloo. The three Waterloo methods tried to extracted updates in two aspects: sentence scoring and event-relevant terms' expansion [53]. The term expansion method investigated by the University of Waterloo is based on bootstrap learning from seed terms. The good results of the three Waterloo methods indicate the effectiveness of the term expansion method, which lead to the best comprehensiveness and latency comprehensiveness. Table 3 shows that our three investigated methods are all above the reported midvalue result, which shows the effectiveness of the three methods. From Table 3, we can see that the KLP method has the best comprehensiveness and latency comp., and KS method has the worst comp. and latency comp., while the performance of SKD method is between the KLP method and KS method. That is to say, comparing with the KS and SKD method, the KLP method utilized a more general metric on scoring updates which can cover much more nuggets.

By comparing the different weights of KLP method from Tables 2 and 3, we can see that the weights on sentence length and sentence position have little effect on the update extraction results in the KLP method. It indicates that the keyword diversity is more important than sentence length and sentence position in the KLP method.

In addition, by combining the results of Tables 2 and 3, we can see that the expected gain has reciprocal relationship with comprehensiveness, like the precision and recall in information retrieval. The KLP method utilizes a more comprehensive metric which considers more factors in scoring sentences. But it is threatened to choose long sentences which leads to the worst gain and latency gain. The KS method proposed only the keyword diversity to evaluate sentences, and it has good performance on expected gain and expected latency gain.

In summary, the keywords in our proposed system are extracted in topic level by using hierarchical LDA. The good results of KS and SKD method, whose key criterion is keywords mentioning in sentence level, show that the SUS extraction is an event- and sentence-level analogue of first topic detection problem. Hence, it is effective when extracted by hierarchical text analysis. Experimental results indicate that a good update should not be a too long sentence which covers many keywords. Generally speaking, the KS method is suitable for systems which have demands on high accuracy, while the KLP method is more suitable for systems which demand high recall.

7. Conclusions

This paper defined the problem of sequential update summarization extraction for unexpected events. To extract relevant timely updates, we formalized a hierarchical sequential update summarization system, which incorporates techniques from topic-level and sentence-level summarization. The hierarchical mining system focused attention on the SUS task and tried to broadcast with useful, new, and timely sentence-length updates about developing unexpected events. To verify the effectiveness of our proposed system, we provided a rounded system based on the SUS task of TREC 2013, including query topics, updates extraction system, and evaluation metrics. We applied the hierarchical update mining system to extract updates of ten unexpected events of the SUS task. Experimental results showed that our proposed system has good performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (NSFC) Grant nos. 61402047 and 61511130081, NSFC Joint Fund with Guangdong under Key Project no. U1201258, Scientific Research Foundation for Returned Scholars, Ministry of Education of China, Sweden STINT Initiation Grant Dnr. IB2015-5959, EU FP7 IRSESMobileCloud Project (Grant no. 612212), and Shandong Natural Science Funds for Distinguished Young Scholar under Grant no. JQ201316. Part of the work presented in this paper has been published in [35].

References

[1] V. Madisetti and A. Bahga, "Internet of things," 2014.

[2] L. Atzori, A. Iera, and G. Morabito, "From 'smart objects' to 'social objects': the next evolutionary step of the internet of things," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, 2014.

[3] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[4] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, "Data mining for internet of things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 77–97, 2014.

[5] Q. Guo, F. Diaz, and E. Yom-Tov, "Updating users about time critical events," in *Advances in Information Retrieval*, pp. 483–494, Springer, 2013.

[6] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: can we trust what we RT?" in *Proceedings of the 1st Workshop on Social Media Analytics (SOMA '10)*, pp. 71–79, ACM, Washington, DC, USA, July 2010.

[7] E. Yom-Tov and F. Diaz, "Out of sight, not out of mind: on the effect of social and physical detachment on information need," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*, pp. 385–394, ACM, Beijing, China, July 2011.

[8] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[9] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, "Summarizing text documents: sentence selection and evaluation metrics," in *Proceedings of the 22nd ACM Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 121–128, Berkeley, Calif, USA, August 1999.

[10] D. Wang, S. Zhu, T. Li, and Y. Gong, "Comparative document summarization via discriminative sentence selection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 3, article 12, 2012.

[11] J. J. Pollock and A. Zamora, "Automatic abstracting research at chemical abstracts service," *Journal of Chemical Information and Computer Sciences*, vol. 15, no. 4, pp. 226–232, 1975.

[12] A. A. A. Esmin, R. S. C. Júnior, W. S. Santos, C. O. Botaro, and T. P. Nobre, "Real-time summarization of scheduled soccer games from twitter stream," in *Natural Language Processing and Information Systems*, E. Métais, M. Roche, and M. Teisseire, Eds., vol. 8455 of *Lecture Notes in Computer Science*, pp. 220–223, Springer, 2014.

[13] A. Patil, K. Pharande, D. Nale, and R. Agrawal, "Automatic text summarization," *International Journal of Computer Applications*, vol. 109, no. 17, pp. 18–19, 2015.

[14] I. Mani and G. Wilson, "Robust temporal processing of news," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 69–76, Association for Computational Linguistics, Stroudsburg, Pa, USA, October 2000.

[15] R. Swan and J. Allan, "Automatic generation of overview timelines," in *Proceedings of the 23rd ACM Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00)*, pp. 49–56, Athens, Greece, July 2000.

[16] W. Ding and C. Chen, "Dynamic topic detection and tracking: a comparison of HDP, C-word, and cocitation methods," *Journal of the Association for Information Science and Technology*, vol. 65, no. 10, pp. 2084–2097, 2014.

[17] M. Osborne, S. Moran, R. McCreadie et al., "Real-time detection, tracking, and monitoring of automatically discovered events in social media," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL '14)*, pp. 37–42, Association for Computational Linguistics, Baltimore, Md, USA, June 2014.

[18] A. Guille and C. Favre, "Mention-anomaly-based event detection and tracking in twitter," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13)*, pp. 375–382, IEEE, Beijing, China, August 2014.

[19] E. Hovy and C.-Y. Lin, "Automated text summarization and the summarist system," in *Proceedings of the TIPSTER Text Program*, pp. 197–214, Association for Computational Linguistics, Baltimore, Md, USA, October 1998.

[20] J. Allan, R. Gupta, and V. Khandelwal, "Temporal summaries of new topics," in *Proceedings of the 24th ACM Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '01)*, pp. 10–18, New Orleans, LA, USA, September 2001.

[21] Temporal summarization, 2013, http://www.trec-ts.org/.

[22] Trec, 2013, http://trec.nist.gov/.

[23] J. Allan, "Introduction to topic detection and tracking," in *Topic Detection and Tracking*, pp. 1–16, Springer, 2002.

[24] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM*, vol. 16, no. 2, pp. 264–285, 1969.

[25] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz, "Multi-document summarization by sentence extraction," in *Proceeding of the NAACL-ANLP Workshop on Automatic Summarization (NAACL-ANLP-AutoSum '00)*, vol. 4, pp. 40–48, Association for Computational Linguistics, April 2000.

[26] M. Georgescu, D. D. Pham, N. Kanhabua, S. Zerr, S. Siersdorfer, and W. Nejdl, "Temporal summarization of event-related updates in wikipedia," in *Proceedings of the 22nd International Conference on World Wide Web (WWW '13)*, pp. 281–284, International World Wide Web Conferences Steering Committee, Rio de Janeiro, Brazil, May 2013.

[27] A. Jatowt and M. Ishizuka, "Temporal web page summarization," in *Web Information Systems—WISE 2004: 5th International Conference on Web Information Systems Engineering, Brisbane, Australia, November 22–24, 2004. Proceedings*, vol. 3306 of *Lecture Notes in Computer Science*, pp. 303–312, Springer, Berlin, Germany, 2004.

[28] H. L. Chieu and Y. K. Lee, "Query based event extraction along a timeline," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 425–432, ACM, July 2004.

[29] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 68–73, ACM, July 1995.

[30] G. Salton, A. Singhal, M. Mitra, and C. Buckley, "Automatic text structuring and summarization," *Information Processing & Management*, vol. 33, no. 2, pp. 193–207, 1997.

[31] C. Zhang, W. Xu, Z. Ma, S. Gao, Q. Li, and J. Guo, "Construction of semantic bootstrapping models for relation extraction," *Knowledge-Based Systems*, vol. 83, pp. 128–137, 2015.

[32] C. Zhang, Y. Zhang, W. Xu, Z. Ma, Y. Leng, and J. Guo, "Mining activation force defined dependency patterns for relation extraction," *Knowledge-Based Systems*, vol. 86, pp. 278–287, 2015.

[33] A. L. Berger and V. O. Mittal, "OCELOT: a system for summarizing Web pages," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00)*, pp. 144–151, ACM, Athens, Greece, July 2000.

[34] G. Erkan and D. R. Radev, "LexRank: graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp. 457–479, 2004.

[35] C. Zhang, Z. Ma, J. Zhang, W. Xu, and J. Guo, "A multi-level system for sequential update summarization," in *Proceedings of the IEEE 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE '15)*, pp. 144–148, Taipei, Taiwan, August 2015.

[36] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[37] C. Zhang, W. Xu, R. Liu et al., "Pris at trec kba," in *Notebook of the TExt Retrieval Conference*, 2013.

[38] Indri, http://www.lemurproject.org/indri.php.

[39] Z. Ma and A. Leijon, "Bayesian estimation of beta mixture models with variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2160–2173, 2011.

[40] Z. Ma, P. K. Rana, J. Taghia, M. Flierl, and A. Leijon, "Bayesian estimation of dirichlet mixture model with variational inference," *Pattern Recognition*, vol. 47, no. 9, pp. 3143–3157, 2014.

[41] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, "Variational bayesian matrix factorization for bounded support data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 876–889, 2015.

[42] T. Minka and J. Lafferty, "Expectation-propagation for the generative aspect model," in *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI '02)*, pp. 352–359, Morgan Kaufmann Publishers, Alberta, Canada, August 2002.

[43] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, supplement 1, pp. 5228–5235, 2004.

[44] "Gibbs lda ++," http://sourceforge.net/projects/gibbslda/.

[45] C. Zhang, Z. Ma, J. Zhang, W. Xu, and J. Guo, "A multi-level system for sequential update summarization," in *Proceedings of the 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE '15)*, Taipei, Taiwan, August 2015.

[46] G. J. Rath, A. Resnick, and T. R. Savage, "The formation of abstracts by the selection of sentences. Part I. Sentence selection by men and machines," *American Documentation*, vol. 12, no. 2, pp. 139–141, 1961.

[47] R. L. Donaway, K. W. Drummey, and L. A. Mather, "A comparison of rankings produced by summarization evaluation measures," in *Proceedings of the NAACL-ANLP 2000 Workshop on Automatic summarization*, pp. 69–78, Seattle, Wash, USA, April 2000.

[48] J. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, and T. Sakai, "Trec 2013 temporal summarization," in *Proceedings of the 22nd Text Retrieval Conference (TREC '13)*, Gaithersburg, Md, USA, November 2013.

[49] "Knowledge based acceleration," 2013, http://trec-kba.org/.

[50] "Kba data," 2013, http://s3.amazonaws.com/aws-publicdatasets/trec/kba/index.html.

[51] Test topics, 2013, http://trec.nist.gov/data/tempsumm/2013/testTopics.xml.

[52] Q. Liu, Y. Liu, D. Wu, and X. Cheng, "ICTNET at temporal summarization track TREC 2013," in *Proceedings of the 22nd Text Retrieval Conference (TREC '13)*, 2013.

[53] G. Baruah, R. Guttikonda, A. Roegiest, and O. Vechtomova, "University of waterloo at the TREC 2013 temporal summarization track," in *Proceedings of the 22nd Text Retrieval Conference (TREC '13)*, Gaithersburg, Md, USA, November 2013.