Information Sciences 646 (2023) 119415

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Efficient graph-based event detection scheme on social media

Kyoungsoo Bok [a], Ina Kim [b], Jongtae Lim [b], Jaesoo Yoo [b,*]

[a] Department of Artificial Intelligence Convergence, Wonkwang University, Iksandae 460, Iksan, Jeonbuk 54538, Korea
[b] Department of Information and Communication Engineering, Chungbuk National University, Chungdae-ro 1, Seowon-Gu, Cheongju, Chungbuk 28644, Korea

ARTICLE INFO

*Keywords:*
Event detection
Social media
Keyword graph
User interest
Betweenness centrality

ABSTRACT

As various opinions and thoughts of users are shared on social media, various events can be detected through social media data analysis. The graph-based event detection scheme may eliminate event duplication detection by clustering words related to an event. In this paper, we propose a graph-based event detection scheme for detecting various events in the real world through social media analysis. The proposed scheme expresses the simultaneous occurrence of words mentioned with an event in graph form. Therefore, it can convey the information about the detected event and avoid duplicate detection. A keyword graph is constructed and keywords related to an event are clustered by analyzing the collected social data. By considering user interest calculated through changes in social activities of users on social media, the proposed scheme can detect event results that receive more responses from users and improve the reliability of the results by excluding indiscriminate advertisements or malicious posts from the results. We assign a weight to the generated keyword graph by considering user interest and calculate an event detection coefficient to determine the event values of the candidate event graphs. We perform various evaluations to demonstrate the superiority of the proposed scheme.

## 1. Introduction

Various social media services are used for expressing and sharing individual or group opinions online. Social media has replaced traditional media and plays a role in conveying information quickly and effectively [2,6,20,22]. Social media platforms such as Facebook, Twitter, and Instagram provide a platform for people to share interests and activities through short posts containing photographs and text [18,19,29]. Twitter is a microblogging service with a broad user group that allows its users to share information through simple messages [11,17,24]. These social media platforms allow users to share opinions and experiences in real time without time and space constraints [34].

As active participation of social users increases, real world experiences and major issues are being provided through social media [1,9,12,13]. In addition, contents generated through various activities, such as sharing and retweeting based on human networks, spreads rapidly through social networks [15,16]. Studies on event detection have been conducted to identify phenomena, events, and accidents in which many people may be interested by analyzing various types of information generated on social media [23,25–27,32]. Recently, complex event detection schemes have been proposed to recognize various situations occurring in real life [35–38]. In [35,36], they proposed complex event detection schemes for detecting semantic events by processing stream data collected from sensors in real time in an IoT environment. In [37], a semantic pooling scheme for event analysis tasks in Internet video was proposed

* Corresponding author.
*E-mail addresses:* ksbok@wku.ac.kr (K. Bok), kimsajangk@naver.com (I. Kim), jtlim@chungbuk.ac.kr (J. Lim), yjs@chungbuk.ac.kr (J. Yoo).

https://doi.org/10.1016/j.ins.2023.119415
Received 25 August 2020; Received in revised form 9 July 2023; Accepted 18 July 2023
Available online 22 July 2023

based on the observation that not all video shots are equally relevant to events of interest to generate discriminative video representation In [38], a bi-level semantical representation analysis scheme was proposed to learn the weights of semantic representations from multiple image/video sources to detect the occurrence of events within video clips. In some extended services linking social media, complex event detection schemes using sensors and video data can be used for event detection. However, since most social media are mostly text-oriented content, there is a limit to directly utilizing existing complex event detection schemes.

Various studies have been conducted to detect events occurring offline through text analysis generated by social media [4,33]. Early keyword-based schemes attempted to identify topics through the appearance frequency of keywords or term frequency-inverse document frequency (TF-IDF) [5,8,10]. In addition, studies on detecting key keywords that are of interest to users at specific times or locations have been conducted [21,28,30]. However, multiple keywords related to the same event are generated; hence, users must determine the relevance of the extracted topics or keywords to identify the meaning of an event. To solve these problems, graph-based schemes have been developed for detecting events by clustering words that appear simultaneously [7,14]. A graph-based event detection scheme can reduce duplicate detection results and efficiently convey event information. In [7], related keywords are constructed in graph form according to an ontology; thus, keywords that are not registered in the ontology cannot be detected as events. In [14], a scheme was proposed for detecting events through clustering by constructing a keyword graph. However, because the number of events to be detected is preset, the precision depends on the number of events. Furthermore, the reliability of the detection results is poor because social activities such as liking, retweeting, and sharing, which express user empathy or sympathy, are not considered [3,11,24].

Keyword-based event detection schemes using topics and core keywords detect a number of generated words associated with events instantaneously because they detect event topics considering the frequency of word appearance. However, since the event detection results are listed as simple keywords, the user must determine the event type or characteristics. Graph-based event detection schemes construct graphs of correlations between words appearing on social media to cluster simultaneously appearing words to derive event results. Therefore, overlapping detection results may be reduced compared to the existing keyword-based event detection schemes, and words related to the event may be provided. However, graph-based event detection schemes using ontology can clarify the semantic representation of events, but there is a limit to detecting only the events contained in the ontology. Graph-based event detection schemes, which do not rely on ontology, cannot guarantee the accuracy of the results because their use intervenes in the clustering process, and cannot eliminate maliciously frequently generated results because they only consider the frequency of word appearance.

In this paper, we propose a scheme for inducing user interest and detecting events through keyword graphs by considering the number of changes in users' social behavior on Twitter to solve the problems of existing graph-based event detection schemes. The proposed scheme constructs a keyword by analyzing posts on social media to detect various keywords that express information regarding an event. The generated keyword graph is clustered using the vertex and edge betweenness centralities shown in the graph. To improve the precision of event detection, we assign user interest and the appearance frequency of keywords as the weights of the graph. We calculate changes in user interest and detect events using changes in the social activities of users. The contributions of this paper are as follows.

- The proposed scheme removes hashtags, special characters, and URLs contained in tweets to improve the accuracy of extraction of related words necessary for event detection, and removes keywords with frequency below threshold.
- Since the proposed scheme performs clustering using the betweenness centrality of vertices and edges, it groups event-related keywords without user intervention.
- The proposed scheme derives user interests by considering changes in social behavior such as 'like' and 'sharing', and weights the line, so it removes results containing advertising posts and malicious content.
- Through event detection based on keyword graphs, the proposed scheme effectively conveys the results by expressing information about the event as a graph expressing the interrelationship of words rather than words.

The remainder of this paper is organized as follows. Section 2 describes the conventional schemes used to detect events and their limitations. Section 3 describes the proposed scheme in detail. In Section 4, the performance evaluation results of conventional event detection schemes are compared with those of the proposed scheme, confirming the superior performance of the proposed scheme. Section 5 concludes the paper, summarizing the findings and suggesting directions for future research.

## 2. Related work

In [28], a scheme was proposed for detecting events by capturing the moment when the emotional state of a user is agitated according to a psychological theory that classifies emotions. Using an emotion model that classifies the emotions of users into seven categories, an emotion dictionary is constructed, and tweets are classified by emotion. Outliers outside the normal distribution are detected. However, it is difficult to accurately classify the emotions of data collected from social media, owing to non-literal expressions such as irony, sarcasm, and dark humor. Furthermore, as the scheme relies solely on emotions, it is impossible to detect events separately when multiple events corresponding to the same emotion occur simultaneously in one area. The same event can cause different emotions, resulting in multiple detections of the same event.

In [30], a scheme was proposed for detecting events using the inverse document frequency (IDF) of the appearance of words. TF-IDF is an algorithm that numerically reflects the importance of a word in a particular document within a set of multiple documents. The TF indicates how often a particular word appears in a document, and the IDF is the multiplicative inverse of the number of times a particular word appears in a document. Thus, a word that does not appear frequently in a document has a high IDF value and can be

used as a keyword. Therefore, when a word that has not often appeared is present in a tweet, an event topic is detected using the changes in the IDF. The scheme using TF-IDF detects event topics by identifying words that have not previously appeared. This scheme can effectively detect event topics with a wide variation range by considering the change rate over time. However, as it derives event keywords as results rather than detecting event related information, it has a shortcoming that the user must infer event information from the listed results.

In [7], a scheme was proposed for constructing a keyword graph and detecting events using word sets and event names classified in a dictionary. The ontology for actors, places, and events is built into a dictionary, and a graph is constructed. After URLs and special characters are eliminated, words that are not included in the ontology consisting of people, places, and event names are removed. Each vertex represents a word included in the ontology. After words that occur together with an event name are connected with edges, the number of simultaneous occurrences is assigned a weight. To create event candidate graphs, the PageRank value of each vertex is calculated. Event candidate graphs are created for the vertices with the highest rank values. The process of concatenating vertices in descending order of weight is repeated until vertices that include information about actors, places, time, and events are obtained. The generated event candidate graphs are derived as detection results. This scheme constructs an event graph based on the ontology and checks whether the event graph includes information that is not related to an event through performance evaluation. However, unregistered words are excluded from the analysis. Moreover, it has a limitation in that it cannot deliver results that are not included in the ontology, even if an event is important and causes major issues.

In [14], a scheme was proposed for detecting events via keyword graph clustering without an ontology. By not using an ontology, the scheme can detect various events and results, in contrast to schemes that detect only restricted words. It performs clustering by receiving the number of events to be detected from users. All vertices are classified into newly appearing words that did not appear previously and important words that consistently appear. The generated graph performs voltage-based clustering [31] by receiving the input of the number of clusters from the user. The changes in the appearance frequencies of all the words in the clustered graph are calculated, and if the changes exceed a threshold value, this is detected as an event.

In [39], Iktishaf+ (an Arabic word meaning discovery) system which is an improved version of Iktishaf [40] was proposed to detect traffic events automatically from tweets using distributed machine learning over Apache Spark. Iktishaf + uses Iktishaf Stemmer for the Arabic language to strip affixes based on the length of the tokens. To generate a training set, approximately twenty thousand tweets were labeled and combined with automatically labeled tweets using the automatic labeling approach. CountVectorizer and IDFModel algorithms provided in the Spark ML generate the feature vectors. The Spark ML library is used to build and train the models. A binary classification is used to avoid assigning two labels to a tweet.

In [41], Embed2Detect was proposed to detect event in social media by combining the characteristics in word embeddings and hierarchical agglomerative clustering (HAC). To facilitate domain, platform or language-independent event detection, self-learned word embeddings are used to generate features of the target corpus. Temporal changes between clusters and vocabularies are taken into account to identify events that do not directly depend on clusters. Cluster change calculation measures nearby word or word group changes over time. After identifying a time window, event word extractor extracts words related to events that occurred in window.

In [42], an online spatiotemporal event detection, which can detect social events in various space–time domains, was proposed. This scheme uses a quad-tree to partition the geographic space into multiple regions based on the density of social media data. It also accurately estimates the event duration by merging events that occur in the same region at consecutive time intervals. To increase the precision of event detection and handle spatiotemporal events occurring over changing resolutions, events are pruned after event merge.

## 3. Graph-based event detection

### 3.1. Overall architecture

Various studies have been performed on the detection of events using social media. The event detection scheme using emotion analysis is fast because it does not have a preprocessing stage [28]. However, it is difficult to classify the emotions of data collected from social media, because of non-literal expressions such as irony. Moreover, as the scheme relies solely on emotions, it is impossible to detect events separately when multiple events indicating the same emotion occur simultaneously in one area. The same event can cause opposing emotions, resulting in multiple detections of the same event. Among the conventional schemes for detecting event topics, a scheme was proposed for detecting events using changes in the IDF of word appearance [30]. This scheme detects words that have not appeared frequently before but have suddenly appeared several times as event topics. However, it produces each event keyword rather than detecting event details. Thus, there is a limitation in that the user must infer event information by reviewing the event topics in the listed results. In [7], a keyword graph is constructed using event names and word ontology to detect events. However, because words that are not in the ontology are excluded during the event detection process, unregistered words can be ignored in the results. Even an event that has caused a significant response from users cannot be detected unless it is included in the ontology. In [14], a scheme was proposed for detecting events using graph clustering without relying on an ontology to solve the problems of previous schemes. The clustering scheme used in [31] has a high processing speed but creates subgraphs depending on the number of clusters desired by the user. Thus, there is a limitation in that the precision is reduced because events that should be separated are merged or the same event is separated by user intervention. Conventional schemes consider the frequency of appearance of words and changes in emotion-related keywords. However, they do not consider social activities, such as liking and sharing, which are features of social media services. Social activities not only indicate the level of user interest but also can spread information to other

users. Therefore, if changes in social activities during event detection are considered, user interest can be reflected. This can improve the reliability of the results, because spam and malicious posts uploaded indiscriminately can be excluded.

In this paper, we propose a graph-based event detection scheme based on graph clustering that considers social activities in social media environments. We construct a keyword graph by analyzing posts on social media and assign user interest as a weight by considering activities such as liking and sharing to improve the reliability of the results. To overcome the limitations of conventional schemes in which the results depend on the number of event detections, we derive candidate graphs via graph filtering and clustering using the vertex and edge betweenness centralities. We detect events by calculating an event detection coefficient considering the user interest in the derived candidate graphs and verify that all the words included in the graphs are actually used together. We provide the user with the top-k event detection results that have undergone verification.

Fig. 1 shows the overall architecture of the proposed event detection scheme. We collect Twitter data for detecting events. During data preprocessing, we eliminate special characters such as URLs, and stop words, extract keywords using a morphological analyzer, and remove keywords that are assumed to occur accidentally. We construct a keyword graph based on the extracted keywords through data preprocessing and perform an event information extraction process in which candidate graphs are generated. To construct a keyword graph, we produce the keywords processed in the previous step with vertices and connect them with edges when keywords occur simultaneously. We assign weights to the edges of the graph after calculating user interest by considering social activities such as liking and sharing. We derive candidate graphs through graph filtering and clustering using the vertex and edge betweenness centralities of the generated initial graph. We perform an event detection process to check whether the derived event are meaningful. We identify events by calculating an event detection coefficient using the edge weights of candidate graphs and finally derive top-k event graphs as results through a verification step for confirming that all words are meaningful.

### 3.2. Data preprocessing

The collected tweets contain information about events but also contain many unnecessary words. For example, they contain hashtags, special characters for using a retweet function, and URLs. Therefore, we perform data preprocessing to obtain the required information from the collected social data. First, we remove unnecessary data for event detection by eliminating special characters, URLs, and stop words. Keywords are extracted from the refined tweets. Finally, we eliminate meaningless data in the results by removing low frequency words.

The nature of social media services often leads many users to post short summaries of messages along with URLs, as these services

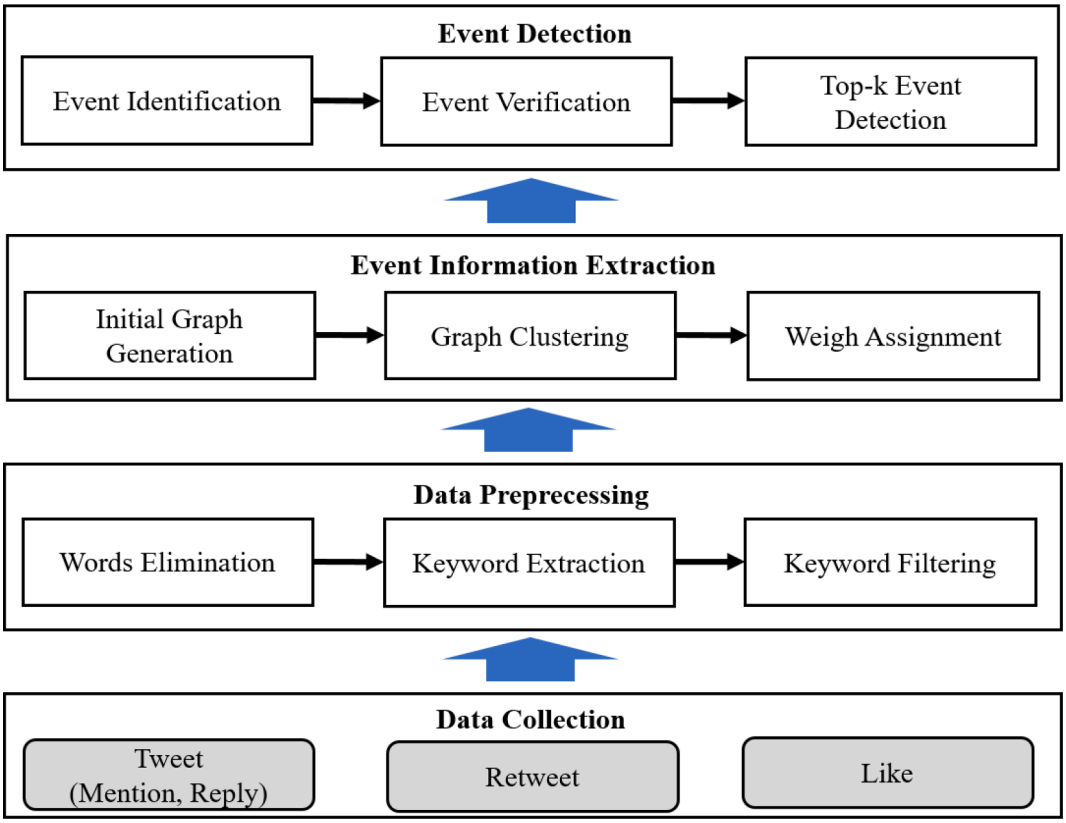



**Fig. 1.** Overall architecture of the proposed event detection scheme.

are unsuitable for posting long messages. In addition, emoticons conveying user emotions—often accompanied by opinion exchanges and special characters such as exclamation marks, periods, and question marks—are used to express nuances. Unnecessary strings for event detection, such as URLs and special characters are eliminated before the algorithm is applied. There are also special characteristics that allow users to use the functions provided by social media services. For example, "Mention" is used in conversations to mention other users, "Retweet" is used to share other users' opinions or add opinions, and hashtags make it easy to index tweets when various people follow tweets on a specific topic. Special characters (e.g., @, #) used in these functions are also eliminated.

After the elimination of special characters, natural language processing is performed for each sentence, to extract the keywords required for analysis. Postpositional particles, which are attached to substantives such as nouns, establish grammatical relations with other words. They are eliminated because they are considered typical stop words. Furthermore, as new words or adverbs are considered not only unimportant elements of the results but also meaningless stop words, they are eliminated. As the final outcomes, the roots of nouns and verbs, which are useful for event detection, are extracted.

Fig. 2 shows an example of the data preprocessing. First, URLs such as "https://omn.kr/oluj" and special characters such as "#" to use functions of social media and "[" are eliminated from the input tweets. Subsequently, postpositional particles and stop words are eliminated. After the stop words are removed, nouns such as "Pohang" (a city in Korea) and "earthquake" are extracted using a keyword extraction function of a morphological analyzer.

Keywords with low frequency may be words that appear by accident and may become meaningless over time. Therefore, keywords with appearance frequencies below the threshold are removed. By removing keywords with a low frequency that account for a large percentage, we increase the accuracy of event detection and reduce the graph generation cost.

### 3.3. Event information extraction

To extract event information, a keyword graph is constructed using the words extracted from the data preprocessing, and clustering is performed. First, an initial graph is generated using preprocessed keywords. Tweets containing event information also include other expressions or ancillary situations that describe the situation. Therefore, it is possible to efficiently extract event information by constructing words that occur together in a graph form. Graph clustering is performed using the vertex and edge betweenness centralities. Here, words appearing with a central keyword that indicates an event are clustered together. Weights are assigned to candidate event clusters with consideration of user interest. User interest is derived from changes in social activities and the appearance frequencies of the keywords. Events can be reliably detected in the event detection step using weights assigned to the

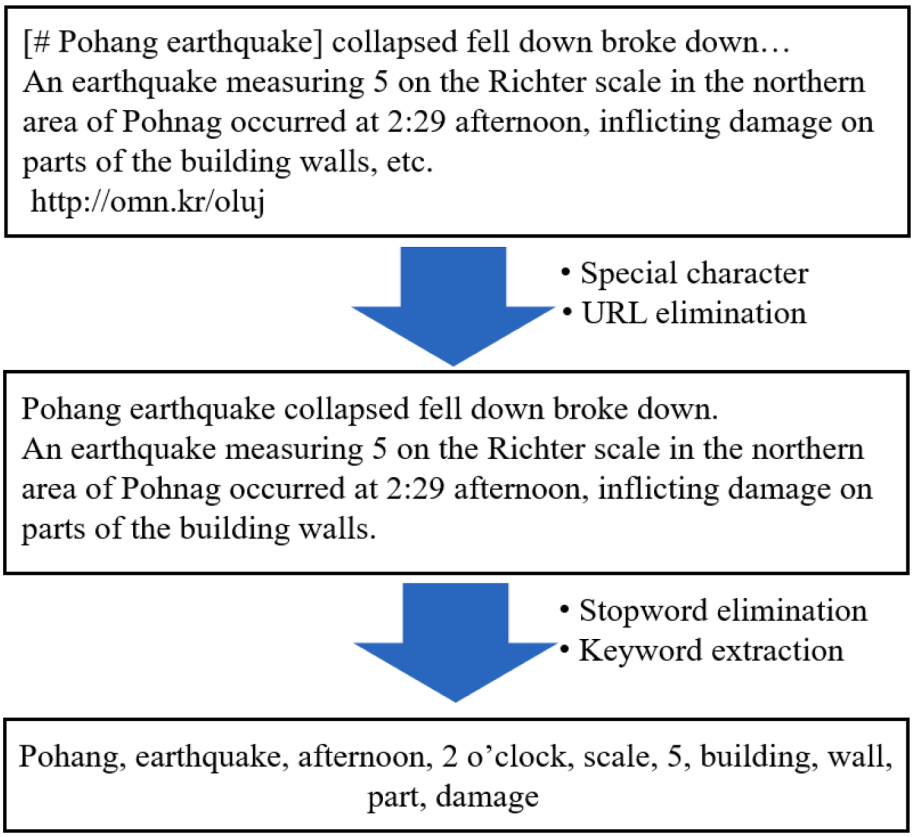



**Fig. 2.** Data preprocessing process.

edges.

An initial graph is generated using keywords that have undergone preprocessing. Words that describe a situation appear together in a tweet that contains event information. Therefore, event information can be efficiently extracted, and duplicate event detection can be avoided by constructing a graph that presents keywords appearing simultaneously. Initial graph generation is performed as follows:

- Set keywords extracted in the preprocessing step as vertices in the graph.
- Connect vertices appearing simultaneously with edges.
- Calculate the appearance frequency and betweenness centrality of each vertex.
- Assign the appearance frequency and betweenness centrality to the attribute values of each vertex.

Suppose that seven keywords are extracted from three tweets. Table 1 represents the vertex ID for extracted keywords. Fig. 3 shows an example of constructing an initial graph, where "cheongju" is a city in Korea. Vertices are created using extracted keywords. In tweets, the vertices representing words that appear together, such as "cheongju" and "flood" or "cheongju" and "evacuation," are connected by adding edges. Each vertex has attribute values obtained by calculating the appearance frequency of a keyword and the vertex betweenness centrality.

A keyword graph $G_t$ based on a window slide interval corresponding to time $t$ is a nondirectional graph with weights. It consists of a set of vertices $V$, a set of edges $E$ connecting vertices, and a set of weights $W$ of edges between the vertices. Each vertex $v_i \in V$ represents a keyword and has the appearance frequency and vertex betweenness centrality $VBC_i$ as attribute values to use in the next clustering step. The vertex betweenness centrality of $v_i \in V$ can be calculated using Eq. (1). $SP_{m,n}$ refers to a set of all shortest paths connecting vertices $v_m$ and $v_n$, and $SP_{m,n}(v_i)$ refers to the number of shortest paths that must pass through vertex $v_i$ among all the shortest paths connecting vertices $v_m$ and $v_n$; that is, for all the shortest paths present in the graph, the vertex betweenness centrality refers to the proportion of the vertices that belong to these paths. Therefore, a high vertex betweenness centrality indicates that the ratio of being mentioned together with other keywords is high, which indicates that the keyword is likely to be a central keyword among the keywords related to an event. Therefore, a central keyword for an event is found using the vertex betweenness centrality in the clustering process, and clustering is performed according to the keyword.

$$VBC_i = \sum_{v_m, v_n \in V} \frac{SP_{m,n}(v_i)}{SP_{m,n}} \tag{1}$$

Each edge $e_i$ connects two vertices that occur multiple times. The weight $w_i$ assigned to edge $e_i$ connecting the two vertices has a user interest value based on the changes in social activities and the frequency of simultaneous occurrence. The assigned weight is used to determine how valuable a particular event is in the event detection step.

Clustering is performed to search for candidate event graphs to be detected in the initially generated keyword graph. The proposed scheme performs graph clustering using vertex and edge betweenness centralities. The proposed scheme overcomes a limitation of conventional schemes, which perform clustering by receiving the number of events to be detected from the user; thus, the shape of the candidate event graph depends on the user request. It performs clustering by finding a central keyword representing an event using the vertex betweenness centrality and then searching and separating other candidate event graphs that are semantically different using the edge betweenness centrality. This process minimizes user intervention and produces consistent clustering results.

Fig. 4 shows the event clustering algorithm of the proposed scheme. The initial graph searches for the central keyword with the highest $VBC_i$ value. Vertices that are more than 2-hops from the central keyword are added to the removal target. This is because they are likely to be unrelated to the central keyword, as they do not occur simultaneously with the central keyword in one sentence. The edge betweenness centrality $EBC_i$ is calculated for all edges in the graph. $EBC_i$ confirms the existence of different event clusters. The edge betweenness centrality of $e_i$ is calculated using Eq. (2). $SP_{m,n}$ refers to a set of all shortest paths connecting vertices $v_m$ and $v_n$, and $SP_{m,n}(e_i)$ refers to the number of shortest paths that must pass through edge $e_i$ among all the edges connecting vertices $v_m$ and $v_n$. Therefore, an edge connecting two event clusters has a high value because it is included in the shortest path. Assuming that the $EBC_i$ of all the edges follows a normal distribution, clustering is performed by dividing the cases into those where there is and is not an edge for which $EBC_i$ is outside the $m+2\sigma$ value. Here, $m$ represents the mean value, and $\sigma$ represents the standard deviation. The edge with the highest $EBC_i$ is cut if it is outside the range. Two vertices connected to the cut edge are copied by creating a new edge for each vertex. This process is repeated from the previous step. Subsequently, if there is no edge outside the range, it is processed by dividing into two cases. First, if the target vertices for removal and the central keyword selected in the first step are in the same cluster, they are removed.

**Table 1**
Vertex ID for extracted keywords.

| Keyword | Vertex ID |
|---|---|
| cheongju | $v_1$ |
| evacuation | $v_2$ |
| preparation | $v_3$ |
| flood | $v_4$ |
| disaster | $v_5$ |
| text | $v_6$ |
| musim river | $v_7$ |

| Tweets | Extracted keywords |
|---|---|
| $tw_1$ | "cheongju", "flood", "preparation" |
| $tw_2$ | "cheongju", "evacuation", "disaster", "text" |
| $tw_3$ | "musim river ", "flood", "cheongju " |

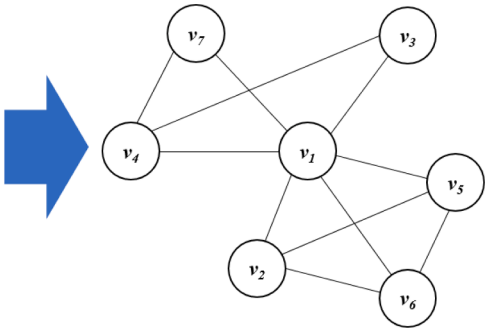



**Fig. 3.** Initial graph construction.

***Algorithm 1. Event Clustering Algorithm***

```
set removalSet, EC ← Ø;
for all v_i ∈ E
   VBC_i ← Edge Betweenness Centrality of v_i;
   topicVertex ← maxVertex(VBC_i);
if distance(topicVertex, j) > 2
   Add vertex j to removalSet;
for all e_i ∈ E
   EBC_i ← Edge Betweenness Centrality of e_i;
   if EBC_i > (m+2σ)
      cut edge e_i with maxEdge(EBC_i) ;
      copy two vertex conntected e_i;
   else
      if topicVertex and removalSet are in same cluster
         remove vertex in removalSet
      else
         set removalSet ← Ø
```

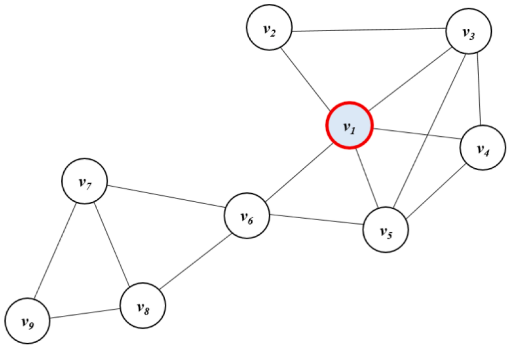
**Fig. 4.** Event clustering algorithm.

Second, if the target vertices for removal and the central keyword selected in the first step are in different clusters, they are released from the removal target. Finally, candidate event graphs are generated by returning each cluster.

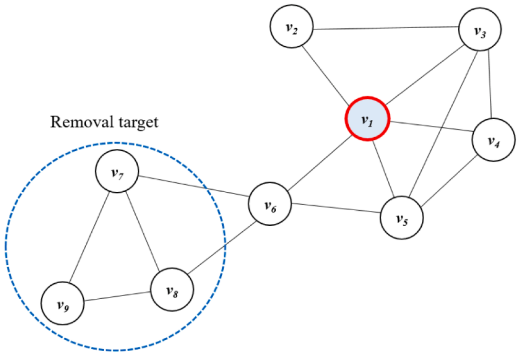
$$EBC_i = \sum_{v_m, v_n \in V} \frac{SP_{m,n}(e_i)}{SP_{m,n}} \tag{2}$$

Fig. 5 shows an example of the graph clustering process of the proposed scheme. It finds for a central keyword vertex with the highest $VBC_i$ in the initially generated graph. In Fig. 5(a), $v_1$, which is marked in a different color, is a vertex. As shown in Fig. 5(b), vertices that are more that 2-hops away from $v_1$ are set as the removal target. Subsequently, the edge betweenness centrality value $EBC_i$ is calculated for all the edges in the graph to confirm the existence of event clusters that differ from each other. This confirms that all $EBC_i$ values are outside the 95% confidence interval. As shown in Fig. 5(c), the edge is cut, as it is outside the confidence interval, and vertices $v_7$ and $v_8$ are copied. This process is then repeated. If there are no more values outside the confidence interval, as the central keyword $v_1$ and removal target are presented in different graphs, as shown in Fig. 5(d), two candidate event graphs are returned.

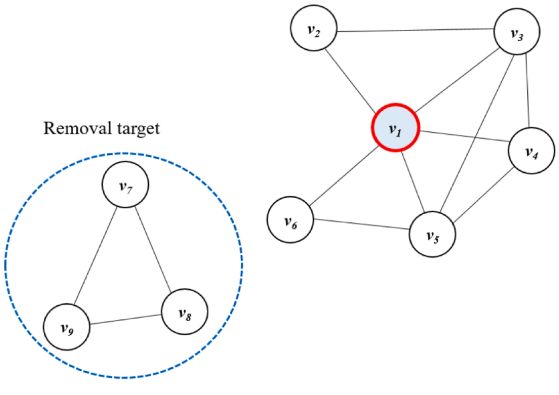
Conventional schemes only consider the appearance frequency of keywords and changes in keywords related to emotions. They do not consider social activities, such as liking and sharing, which are functions provided in social media environments. Social activities



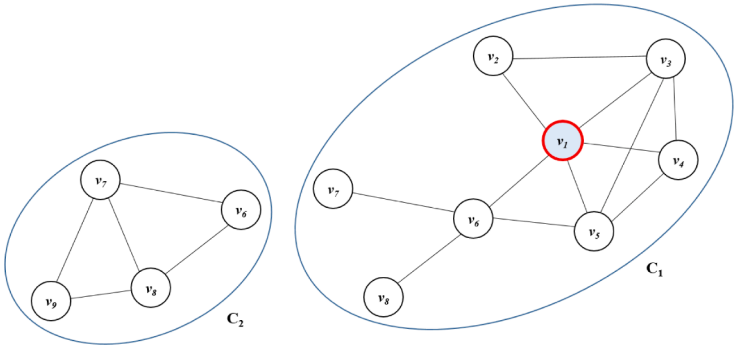

(a) Initial graph.




(b) Removal target Selection.

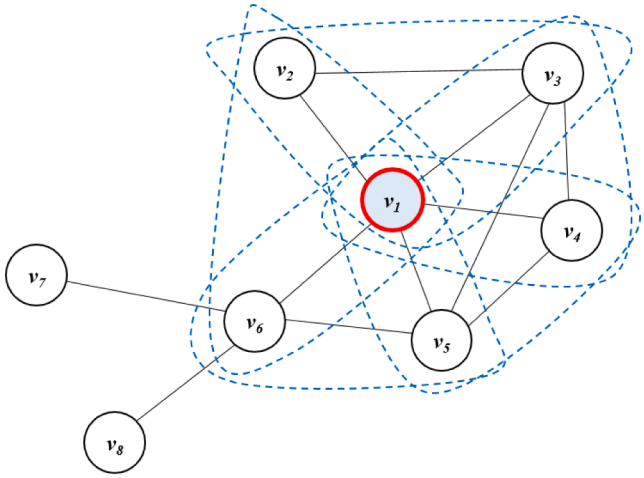



(c) Edge cut.

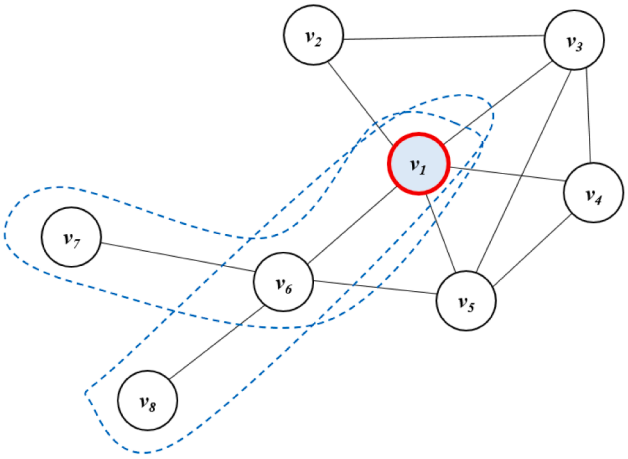



(d) Candidate event graph.

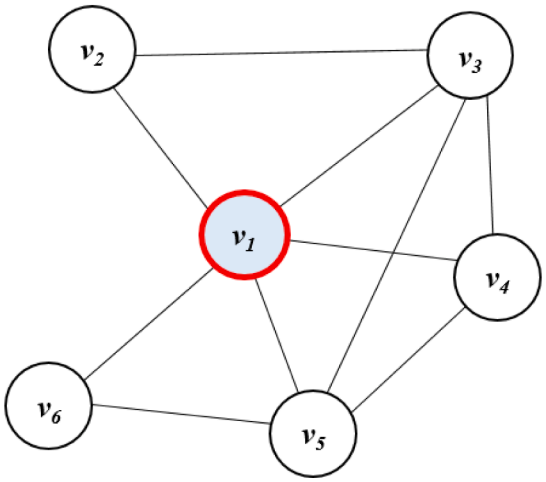
**Fig. 5.** Example of the clustering process.




(a) Verification for the 1-hop range.




(b) Verification for a range of more than 2-hops.




(c) Results after verification.

**Fig. 6.** Example of the event verification process.

can not only spread information to other users but also express user interest. Therefore, in event detection, user interest can be reflected by considering changes in social activities, and the reliability of the results can be improved by excluding indiscriminate spam or malicious posts from the results.

User interest is calculated using changes in social activities and the frequency of the simultaneous appearance of two keywords and is assigned as a weight to edges connecting two keywords in the event graph. It is derived using changes in the social activities of tweets where two keywords appear simultaneously and the frequency of the simultaneous appearance of two keywords. It is assigned as weights to the edges. Weights are used in event detection to determine whether an event is meaningful.

According to Eq. (3), user interest is calculated via the changes in the user social activities of posts where two words appear simultaneously. Changes in the number of retweets and likes are reflected in the formula. $RTN_t(k_i, k_j)$ and $LN_t(k_i, k_j)$ represent the numbers of "retweets" and "likes," respectively, of a tweet where keywords $k_i$ and $k_j$ appear simultaneously at time $t$. $RTN_{t-1}(k_i, k_j)$ and $LN_{t-1}(k_i, k_j)$ represent the numbers of "retweets" and "likes," respectively, of a tweet where keywords $k_i$ and $k_j$ appear simultaneously at time $t-1$. To normalize the changes calculated via Eq. (3) and adjust the weight to be assigned depending on the change rate, Eq. (4) is applied. Here, $\mu$ and $\beta$ are arbitrary constants for adjusting the weight range, and $S_{ij}$ is a value calculated using Eq. (3).

$$S_{ij} = \frac{RTN_t(k_i, k_j) + LN_t(k_i, k_j)}{RTN_{t-1}(k_i, k_j) + LN_{t-1}(k_i, k_j)} \tag{3}$$

$$NS_{ij} = \mu \times (\beta^{S_{ij}} - 1) \tag{4}$$

Eq. (5) assigns weights using the frequency of the simultaneous appearance of the two words. $CF(k_i, k_j)$ refers to the frequency of the simultaneous appearance of keywords $k_i$ and $k_j$.

$$F_{ij} = logCF(k_i, k_j) \tag{5}$$

Finally, Eq. (6) assigns weights to edges according to the user interest and the frequency of tweets where two words appear simultaneously. $\alpha$ is an arbitrary constant that can be adjusted depending on the user interest and the distribution of the appearance frequency of the two keywords. $NS_{ij}$ and $F_{ij}$ are calculated using Eqs. (4) and (5), respectively, and refer to the weights based on the changes in social activities and frequency. Weights are assigned to all the edges of the candidate event graphs using Eq. (5), and the assigned weights are used for event detection.

$$w_{ij} = \alpha NS_{ij} + (1 - \alpha)F_{ij} \tag{6}$$

### 3.4. Event detection

The $n$ candidate event graphs include various types of event content. Among them, event graphs with high user interest and frequent occurrences should be detected primarily as a result. It is better to not provide results that are worthless as an event. Therefore, a process is needed to identify values as real events for candidate event graphs. In the event identification step, a graph having many issues according to the user's level of interest is detected as a result using weights calculated from the candidate event graph. Event identification is used to determine the event values of the generated candidate event graphs. The event detection coefficient $DC_i$ of event graph $EG_i$ is calculated to determine the event value of the candidate event graph. After ranking the results in descending order of $DC_i$, the top-k event graphs requested by the users are detected.

The event detection coefficient $DC_i$ of the event graph $EG_i$ is calculated for each candidate event graph. $DC_i$ is calculated using Eq. (7). If $v_m$ and $v_n$ refer to vertices in the candidate event graph, $w_{mn}$ refers to the weight of an edge connecting these vertices. $DC_i$ is the sum of the weights of the candidate event graph considering the user interest and frequency and represents the user interest in the words included in the graph. The higher $DC_i$ value indicates more frequent occurrence and interest from more users. The $DC_i$ of each candidate event graph is calculated, and the top-k event graphs requested by the users are derived as a result. The $DC_i$ values calculated using Eq. (7) are returned in descending order according to the value requested by the users.

$$DC_i = log \sum_{v_m, v_n \in V} w_{mn} \tag{7}$$

Keywords that are unnecessarily added or outside the subject can be included in the clustering process. Therefore, a verification step is required before detecting the event graphs derived as a result. By performing the verification step, high precision event detection results can be obtained. A central keyword with the highest $VBC_i$ is searched among the vertices in the candidate event graph. Among the vertices within 1-hop from the central keyword, two vertices are paired to check whether they occur simultaneously in a sentence. If among the vertices within 1-hop, a word does not occur simultaneously with other words above the threshold value λ, it is considered to be outside the subject of the event graph or to be an added vertex in the clustering process; thus, the word is removed. For vertices more than 2-hops away, we check whether a vertex simultaneously occurs with other vertices located on the shortest path from the corresponding vertex to the central keyword. If it does not simultaneously occur with words above the threshold value, we derive the final event graph as a result after removing it.

Fig. 6 shows an example of the event verification process. A central keyword for the detected candidate event graph is first identified, and the central keyword and the vertices within 1-hop from the central keyword are paired to check whether they occur simultaneously, as shown in Fig. 6(a). Subsequently, the final event graph is produced by removing vertices more than 2-hops from the central keyword through the verification step. The vertices within 1-hop are paired based on the central keyword $v_1$. As keyword pairs

above the threshold occur simultaneously, the corresponding vertex is maintained. The same process is performed for all words. As shown in Fig. 6(b), vertices in the range of 2 or more hops are verified. We search for the shortest path from word $v_7$ to central keyword $v_1$ and confirms whether $v_6$ and $v_1$ included in the shortest path occur simultaneously. Here, vertices that do not occur simultaneously with $v_1$ and do not satisfy the condition are removed. The results are shown in Fig. 6(c).

## 4. Performance evaluation

Although various schemes have recently been proposed to detect events on social media, they extract word-centered key keywords or detect events around changes in keywords over time [39,41,42]. Therefore, it is impossible to express the association of words associated with the event, thereby detecting duplicate events. In [7], pre-classified event names and words are formed into a keyword graph and events are detected through ontology for actors, places, and events. In [14], a graph-based event detection scheme that performs clustering according to the number of events input to the user was proposed. In [30], a traditional keyword-based event detection scheme was proposed by using the amount of change in IDF to detect event topics when words that do not usually appear frequently appear in tweets. We selected the schemes proposed in [7,14], which are recent graph-based event detection similar to the proposed scheme as performance comparison targets. In addition, for performance comparison with the traditional word-centered event detection, the event detection scheme using TF-IDF[30] was selected for performance comparison. We named the schemes proposed in [7,14,30] as OBED [7], EDO [14], and IDFED [30], respectively.

Table 2 shows the experimental environment. We collected tweets, the number of retweets, and the number of likes using the Twitter4j API, and used them for performance evaluation. Table 3 shows the datasets used in the performance evaluations. We used dataset A with 902,165 tweets collected from 09/01/2017 to 09/30/2017, dataset B with 1,102,534 tweets collected from 10/01/2017 to 10/31/2017, and dataset C with 927,733 tweets collected from 11/01/2017 to 11/30/2017.

The proposed scheme gives the user's social behavior change as a weight of the edge in the process of constructing a graph for keywords extracted from Twitter. In addition, when weighting edges, we determine which attribute to consider more, user interest or appearance frequency of words, through $\alpha$. In the event detection step, verification is performed on the event extracted through $\lambda$. For the ablation study of the proposed scheme, we perform our own performance evaluation of the characteristics considered in the proposed scheme. Self-performance evaluation compares the performance of event detection results according to weighting of edges with event detection results according to $\alpha$ and $\lambda$. We compare the performances of the proposed and existing schemes in terms of processing time, accuracy, reproduction rate, F-measure, and DE-RATE. To compare the performance of the proposed scheme with existing schemes, we evaluated the processing time and accuracy of the results. For the accuracy evaluation, the precision, recall, F-measure, and duplicate event rate (DE-RATE) were compared. Eqs. (8)–(11) give the precision, recall, F-measure, and DE-RATE, respectively, where $N_e$ represents the number of detected events, $N_r$ represents the number of real events, $N_w$ represents the total number of detected events, and $N_d$ represents the number of duplicate detected events.

$$Precision = \frac{N_r \cap N_e}{N_e} \tag{8}$$

$$Recall = \frac{N_r \cap N_e}{N_r} \tag{9}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precison + Recall} \tag{10}$$

$$DE - RATE = \frac{N_d}{N_w} \tag{11}$$

The proposed scheme assigns a weight to an edge by considering user interest. The assigned weight plays an important role in the event identification step. The effect of user interest reflection on event detection was evaluated. We used the event detected between 14:00 and 15:00 on 10/25/2017. Table 4 presents the event detection results that reflect only the appearance frequency of keywords, whereas Table 5 presents the event detection results of the proposed scheme, wherein user interest was considered. In Table 4, the detected event ranked second was messages posted continuously for promotion purposes. There were numerous tweets, but user reactions did not translate into social activities; thus, they were not detected as an event in the proposed scheme. In Table 5, the detected events ranked third and fifth were detected as events because of the rapid increase in user social activities and the number of mentions. The difference in the detected event results indicates that events that receive more empathy and trust from users can be promoted to a

**Table 2**
Performance evaluation environment.

| Feature | Component |
|---|---|
| Processor | Intel(R) Core(TM) i5-6500 3.20 GHz |
| Memory | 8.0 GB |
| OS | Windows 7 Ultimate K 64bit |
| Language | Java 1.8.1, Python 3 |
| DBMS | MongoDB 3.6.3 |

**Table 3**
Datasets used for performance evaluation.

| Dataset | Period | Size |
|---|---|---|
| Dataset A | 2017.09.01 ~ 2017.09.30 | 902,165 |
| Dataset B | 2017.10.01 ~ 2017.10.31 | 1,102,534 |
| Dataset C | 2017.11.01 ~ 2017.11.30 | 927,733 |

**Table 4**
Top five detection results obtained without consideration of user interest.

| Rank | Event |
|---|---|
| 1 | Young-hak Lee, Stepfather, Suicide note, Suicide, Molar, Home, Yeongwol |
| 2 | Taek-jin Kim, Advertising, Baseball, Lineage, NC, Coupon |
| 3 | Man U, Swansea, Sung-yong Ki, Crowd, Full time, Multi, Wales |
| 4 | Rain, Tae-hee Kim, Birth, Daughter, Parent, Baby girl, News, Unit |
| 5 | Park, State-appointed, Attorney, Close-door, Selection, Designation |

**Table 5**
Top five detection results of the proposed scheme.

| Rank | Event |
|---|---|
| 1 | Young-hak Lee, Stepfather, Suicide note, Suicide, Molar, Home, Yeongwol |
| 2 | Rain, Tae-hee Kim, Birth, Daughter, Parent, Baby girl, News, Unit |
| 3 | Culture, Day, Performance, Movie, Event, Art museum, Heize |
| 4 | Geun-hye Park, State-appointed, Attorney, Close-door, Selection, Designation |
| 5 | Thor, Ragnarok, Opening, Marvel, Good review, Anticipation, Box office |

higher rank and detected as events.

Figs. 7 and 8 show the changes in the event detection coefficient from 7:00 to 19:00 for events related to "Culture day" and "Thor". For the proposed scheme, the $DC_i$ value increased owing to the surge in user social activities, and the event detection results are given the upper rank. However, the scheme considering only the appearance frequency of a keyword does not consider the social behavior, so the event cannot be detected.

Fig. 9 shows the changes in the $DC_i$ value from 7:00 to 19:00 on the event graph related to "Lineage." If only the appearance frequency of a keyword was considered for event detection, the change of the $DC_i$ value is large. Therefore, posts for promotional purposes are detected as events. There were numerous tweets, but user reactions did not translate into social activities. Thus, there were no changes in social activities, and it was not detected as an event in the proposed scheme because there was no change in the $DC_i$ value.

The proposed scheme assigns the weight of an edge by considering user interest. By adjusting $\alpha$ in the formula for assigning weights, the attribute that is more weighted between user interest and the appearance frequency of words can be set. We conduct performance evaluations according to the change of the weight by adjusting $\alpha$, and compares the accuracy of the detection results according to the value. Fig. 10 shows the result of conducting self-performance evaluations on the $\alpha$ value. The measurement result shows that the precision, recall, and F-measure were improved on average by 15%, 21%, and 18%, respectively, when$\alpha = 9$. The performance of event

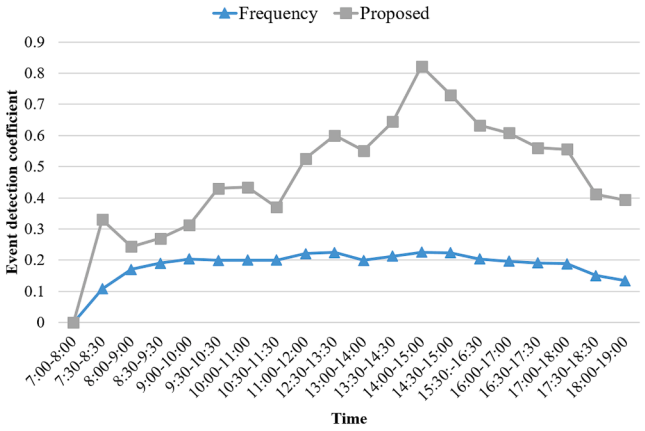



**Fig. 7.** Trend of the event detection coefficient of "Culture day.".

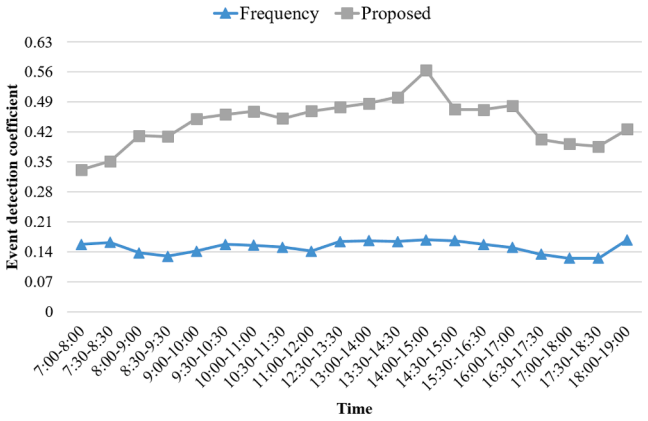



**Fig. 8.** Trend of the event detection coefficient of "Thor.".

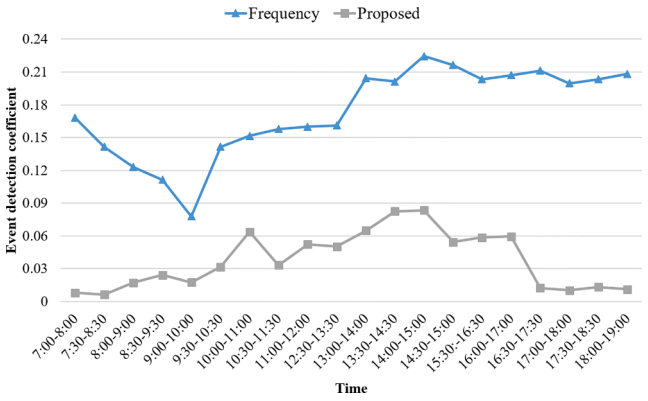


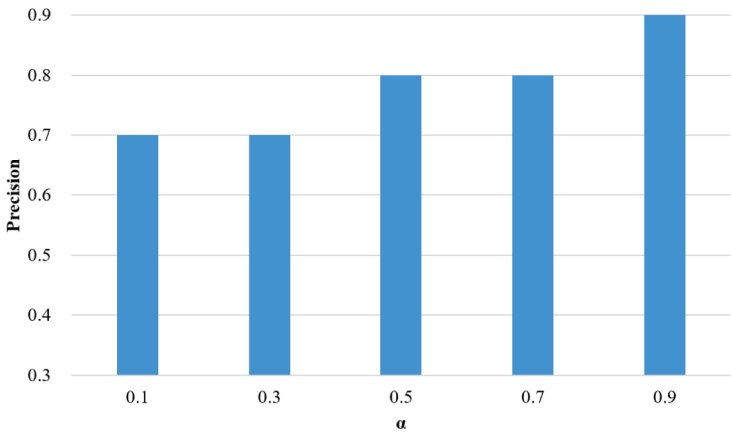

**Fig. 9.** Trend of the event detection coefficient of "Lineage".

detection results is improved by assigning a higher weight to user interest. Based on this result, we used the weight of$\alpha = 0.9$ for comparative evaluations with conventional schemes.

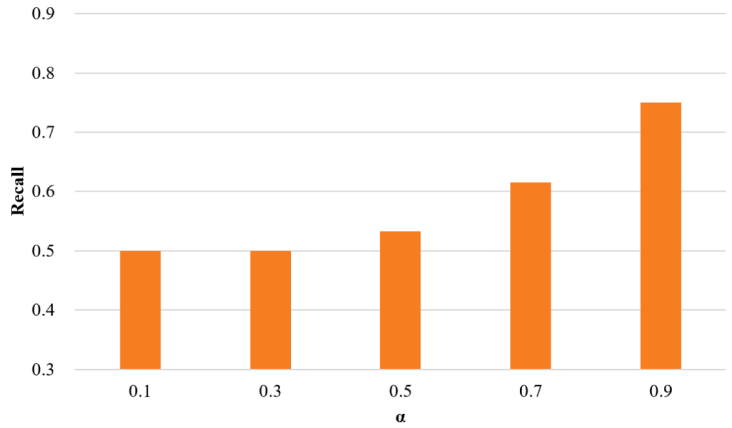
At the last step of event detection in the proposed scheme, the verification is performed to remove inappropriate words for events. The efficiency of verification can be controlled by adjusting the λ value during the verification process. We conducted experimental evaluations according to the change in λ and compared the purity of the detection results according to λ. Purity indicates the degree to which an event detection result does not include words related to other events. That is, it is set as the evaluation metric of the verification step because it determines whether only the desired result is detected without mixing it with other event contents. Fig. 11 shows a graph showing the purity according to λ and the average number of words included in the event graph for dataset A. Of the top 20 results detected in dataset A, the number of events included in each detection result was measured. The more cases where an event is included in an event result, the higher is the purity. Measurement results show that the larger the λ value, the higher is the purity. However, when a large λ value is set, the purity increases but the number of words included in the event graph sharply decreases, as words that can describe the event are removed. That is, a good result is to derive an appropriate result with high purity. Therefore, we used a weight of $\lambda = 0.4$, including more than 15 words with the purity of more than 50%. Based on this result, we set λ to 0.4 for comparative evaluations with conventional schemes and evaluated their performances.

Processing time is an important factor in event detection schemes that analyze large amounts of social data. While changing the size of dataset, we measure the time required for event detection of the proposed scheme and the conventional schemes OBED [7] and EDO [14]. As IDFED [30] does not include the clustering process, in contrast to the proposed scheme, we excluded it from the comparative evaluations of the processing time. Fig. 12 shows the results of the comparative evaluations of the event detection processing time. Although all schemes showed good overall processing performance when processing a small amount of data, the relative processing time increases as the amount of data increase. In OBED, the processing time increases rapidly because the number of ontology comparisons increases as the amount of data increases. The proposed scheme and EDO exhibited moderate increases in the processing time with an increase in the amount of data. Because the proposed scheme reduced the time spent compared with the ontology, it outperformed OBED by approximately 10% with regard to processing time.

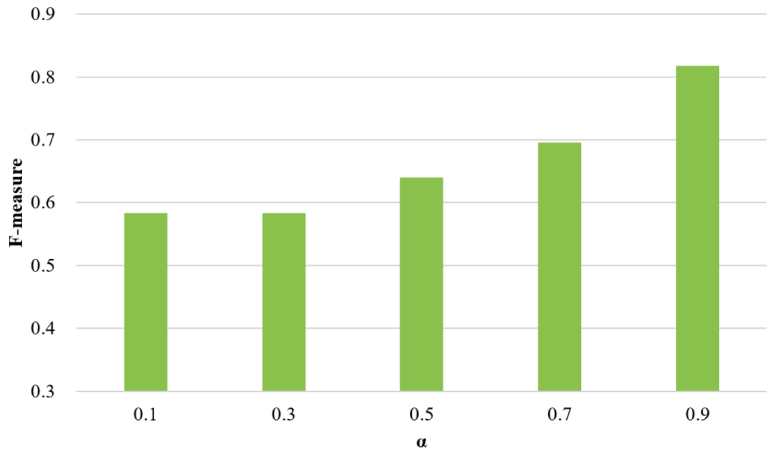
Fig. 13 shows the precision for the proposed scheme and conventional schemes based on the event results detected using datasets A and B. We measured the precision when k (the number of events to be detected) was set to 20, 25, 30, 35, 40, 45, and 50 and compared




(a) Precision




(b) Recall




(c) F-measure

**Fig. 10.** Self performance evaluation with different values of $\alpha$.

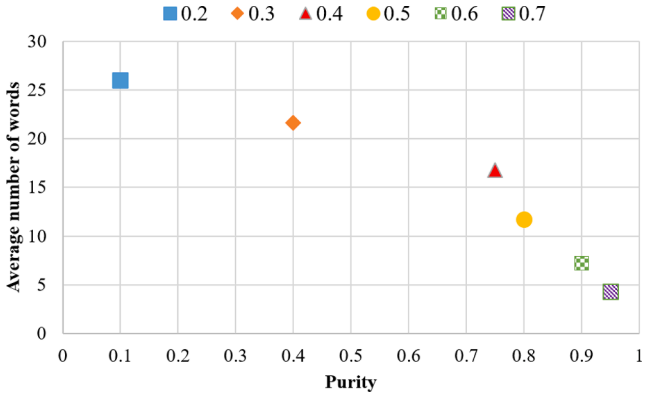



**Fig. 11.** Purity according to λ and the average number of words included in an event graph.

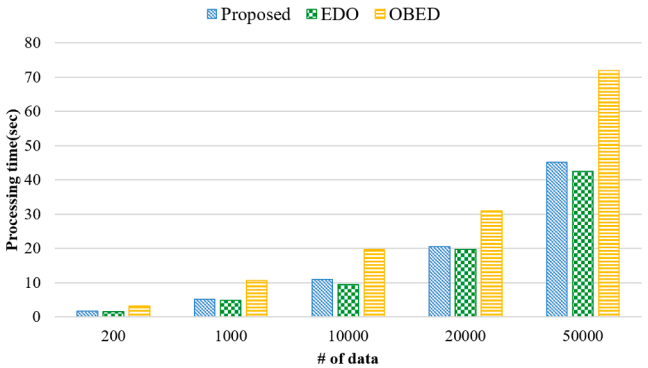



**Fig. 12.** Processing time.

the average precision. The measurement results indicated that the proposed scheme outperformed the conventional schemes. The average performance was 14% higher than that of the conventional schemes. The OBED did not detect events that were not registered in the ontology, resulting in low overall precision. For EDO, as the number of detected events increased, the precision rapidly decreased, and the performance was degraded. Because the clustering results depend on the value of k, an event graph that should be detected as one may be separated.

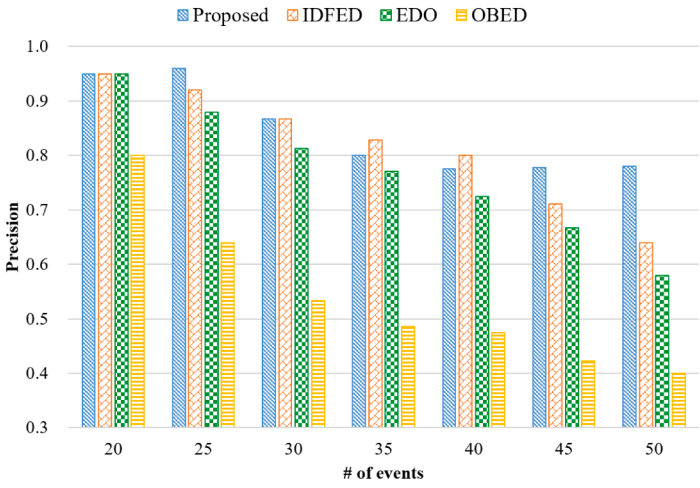
Fig. 14 shows the recall for the proposed scheme and conventional schemes based on the event results detected using datasets A, B, and C. We compared the average recall while changing k and the number of events to be detected. The proposed scheme had an average recall 10% higher than that of the conventional schemes.

Fig. 15 shows the F-measure for the proposed scheme and conventional schemes based on the event results detected using datasets A, B, and C. We compared the average F-measure while changing k and the number of events to be detected. The comparative evaluation results indicated that the F-measure improved by an average of 11.5% for the proposed scheme compared with the conventional scheme. The conventional schemes lacked diversity of results owing to their dependence on the ontology, or their precision and recall were reduced because of user intervention. In comparison, the proposed scheme had better the precision, recall, and F-measure, as it detected events using an automated techniques that considered user interest.

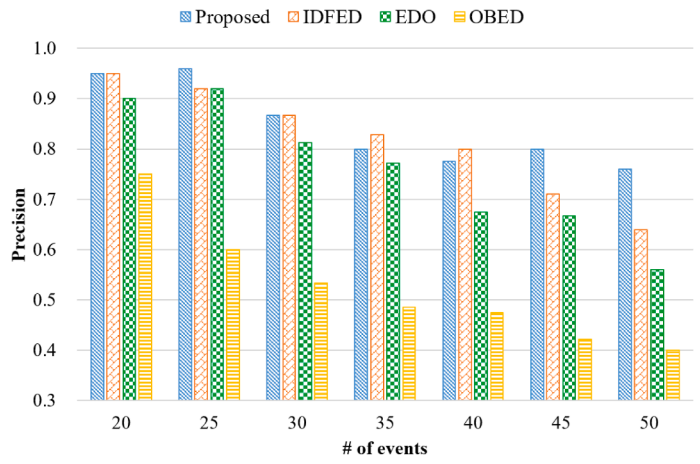
Fig. 16 shows the results of comparative evaluations of the DE-RATE between the proposed scheme and conventional schemes based on the event results detected using datasets A, B, and C. We measured the DE-RATE while changing k and the number of events to be detected. Compared with EDO and IDFED, the DE-RATE of the proposed scheme was reduced by 14% on average. The conventional schemes detected more duplicate events with an increase in the number of events to be detected. IDFED achieved good precision, but the word-level results caused a high DE-RATE. As more event topics were detected as results, the DE-RATE increased, because various words representing an event were given as results. For EDO, as the clustering results varied depending on the value of k, an event graph that should have been detected as one may have been separated, resulting in a high DE-RATE.

## 5. Conclusion

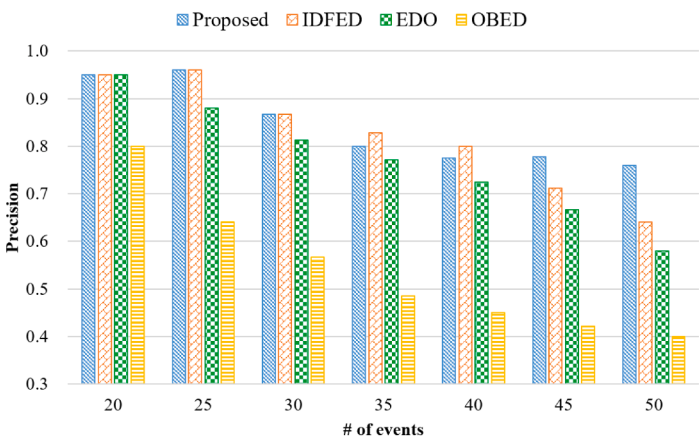
In this paper, we proposed a graph-based event detection scheme that considers user interest in social media environments. We constructed a keyword graph using collected social data. We derived candidate graphs via graph filtering and clustering using the vertex and edge betweenness centralities of the constructed keyword graph. We assigned a weight to the edges of the candidate graphs




(a) Dataset A

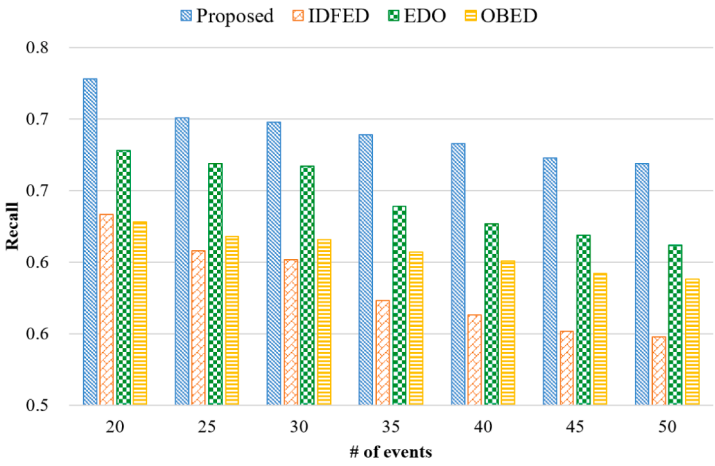



(b) Dataset B

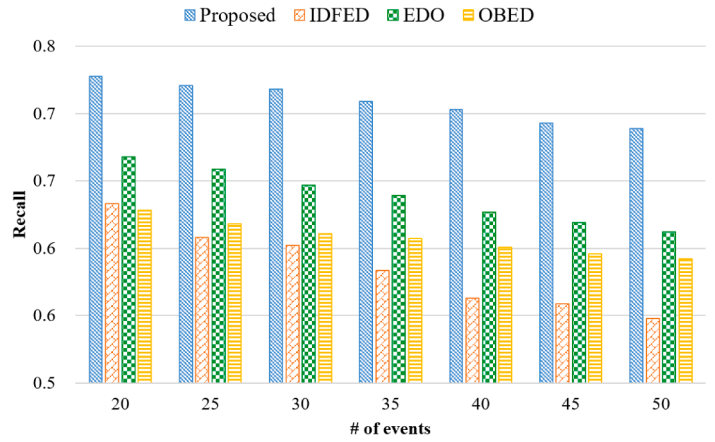



(c) Dataset C

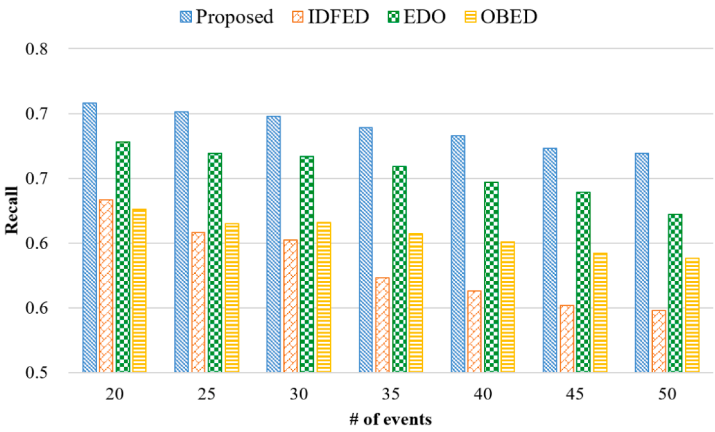
**Fig. 13.** Precision.




(a) Dataset A

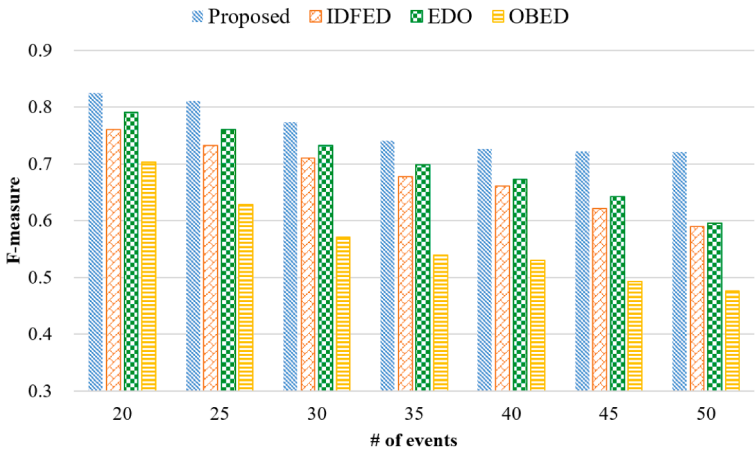



(b) Dataset B

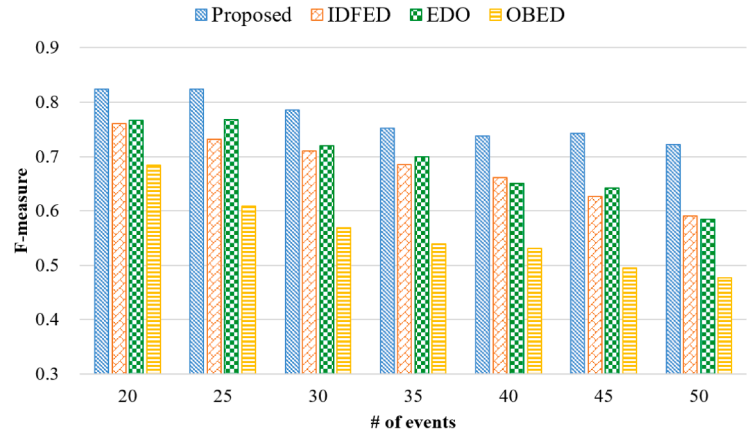



(c) Dataset C

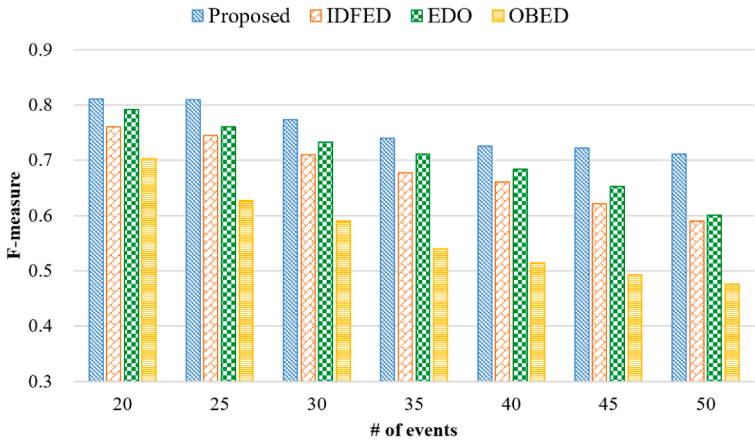
**Fig. 14.** Recall.




(a) Dataset A

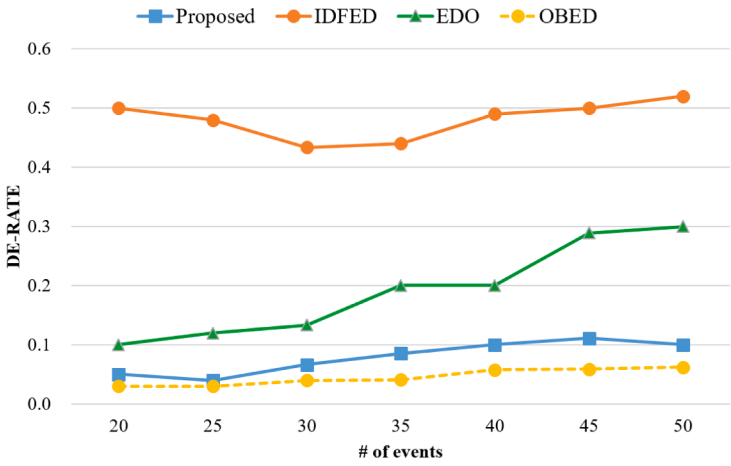



(b) Dataset B

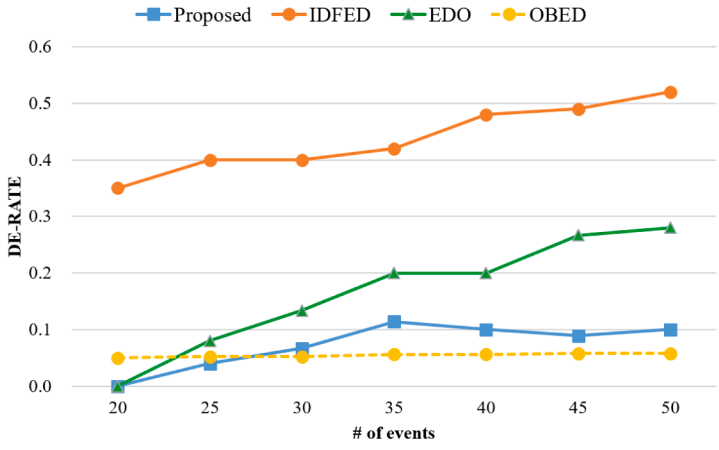



(c) Dataset C

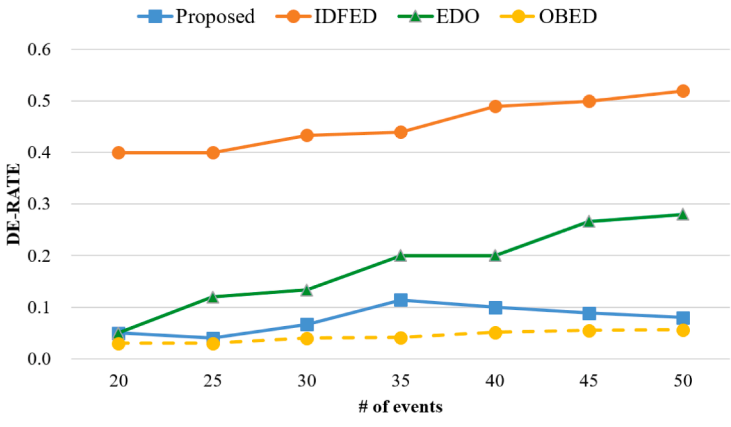
**Fig. 15.** F-measure.




(a) Dataset A




(b) Dataset B




(c) Dataset C

**Fig. 16.** DE-RATE.

by calculating the user interest. The final events were derived through a verification step to confirm the simultaneous occurrence of words after determining whether an event was meaningful using the weight of the edges that reflected user interest. Compared with conventional schemes, the precision, recall, and F-measure of the proposed scheme were improved by 14%, 9%, and 11%, respectively. Additionally, the proposed scheme had a lower DE-RATE than the conventional schemes, indicating its excellent performance. In the future, we will conduct performance evaluations by applying the proposed scheme to data from various social media services, as well as data related to disasters and infectious diseases. Most social media use text content, but recently, extended services including sensor data and multimedia content have been used in social Internet of things and social vehicle networks. In a future study, we will apply the complex event processing scheme for application in social media services utilizing sensors and multimedia content. In addition, we will conduct research to use it in areas that can provide information such as disasters and infectious diseases.

## CRediT authorship contribution statement

**Kyoungsoo Bok:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Ina Kim:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Jongtae Lim:** Methodology, Software, Validation, Data curation. **Jaesoo Yoo:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT). (No. 2022R1A2B5B02002456, RS-2023-00245650), by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2014-3-00123, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis), and by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2023-2020-0-01462) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation).

## References

[1] A. Abdulrahman, J. Lee, Event detection on large social media using temporal analysis, in: Proc. Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 2017, pp. 1-6.
[2] Y.A. Ahmed, M.N. Ahmad, N. Ahmad, N.H. Zakaria, Social media for knowledge-sharing: A systematic literature review, Telematics Inform. 37 (2019) 72–112.
[3] F. Atefeh, W. Khreich, A survey of techniques for event detection in Twitter, Comput. Intell. 31 (1) (2015) 132–164.
[4] P.H. Barros, I. Cardoso-Pereira, H. Allende-Cid, O.A. Rosso, H.S. Ramos, Leveraging phase transition of topics for event detection in social media, IEEE Access 8 (2020) 70505–70518.
[5] K. Bok, Y. Noh, J. Lim, J. Yoo, Hot topic prediction considering influence and expertise in social media, Electron. Commer. Res. 21 (3) (2021) 671–687.
[6] C.T. Carr, R.A. Hayes, Social media: Defining, developing, and divining, Atlantic J. Commun. 23 (1) (2015) 46–65.
[7] A. Edouard, E. Cabrio, S. Tonelli, N.L. Thanh, Graph-based event extraction from twitter, in: Proc. International Conference Recent Advances in Natural Language Processing, Varna, Bulgaria, 2017, pp. 222-230.
[8] Y. Endo, H. Toda, Y. Koike, What's hot in the theme: Query dependent emerging topic extraction from social streams, in: Proc. International Conference on World Wide Web Companion, Florence, Italy, 2015, pp. 31-32.
[9] P. Guan, J. Wu, Effective data communication based on social community in social opportunistic networks, IEEE Access 7 (2019) 12405–12414.
[10] J. Guzman, B. Poblete, On-line relevant anomaly detection in the Twitter stream: an efficient bursty keyword detection model, Chicago, Illinois, USA, Proc. ACM SIGKDD Workshop on Outlier Detection and Description, 2013, pp. 31–39.
[11] M. Hasan, M.A. Orgun, R. Schwitter, A survey on real-time event detection from the Twitter data stream, J. Inf. Sci. 44 (4) (2018) 443–463.
[12] T. Hua, N. Yue, F. Chen, C. Lu, N. Ramakrishnan, Topical analysis of interactions between news and social media, in: Proc. AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA, 2016, pp. 2964-2971.
[13] T. Hua, L. Zhao, F. Chen, C.-T. Lu, N. Ramakrishnan, How events unfold: Spatiotemporal mining in social media, ACM SIGSPATIAL Special 7 (3) (2016) 19–25.
[14] S. Katragadda, R. Benton, V. Raghavan, Framework for real-time event detection using multiple social media sources, in: Proc. Hawaii International Conference on System Sciences, Hilton Waikoloa Village, Hawaii, USA, 2017, pp. 1716–1725.
[15] W. Kaur, V. Balakrishnan, O. Rana, A. Sinniah, Liking, sharing, commenting and reacting on Facebook: User behaviors' impact on sentiment intensity, Telematics Inform. 39 (2019) 25–36.
[16] K. Keib, I. Himelboim, J. Han, Important tweets matter: Predicting retweets in the #BlackLivesMatter talk on twitter, Comput. Hum. Behav. 85 (2018) 106–115.
[17] M. Kosugi, K. Utsu, S. Tajima, M. Tornita, Y. Kajita, Y. Yamamoto, O. Uchida, Improvement of Twitter-based disaster-related information sharing system, in: Proc. International Conference on Information and Communication Technologies for Disaster Management, Münster, Germany, 2017, pp. 1–7.
[18] H. Kwak, C. Lee, H. Park, S. Moon, What is Twitter, a social network or a news media? In: Proc. International Conference on World Wide Web, Raleigh, North Carolina, USA, 2010, pp. 591-600.
[19] C.S. Lee, N.A.B.A. Bakar, R.B.M. Dahri, S.J. Sin, Instagram this! Sharing photos on Instagram, Proc. In: International Conference on Asia-Pacific Digital Libraries, Seoul, Korea, 2015, pp. 132-141.

[20] Y. Liu, T. Bakici, Enterprise social media usage: The motives and the moderating role of public social media experience, Comput. Hum. Behav. 101 (2019) 163–172.
[21] M. Mathioudakis, N. Koudas, Twittermonitor: Trend detection over the twitter stream, in: Proc. ACM SIGMOD International Conference on Management of Data, Indianapolis, Indiana, USA, 2010, pp. 1155-1158.
[22] S. Oh, S.Y. Syn, Motivations for sharing information and social support in social media: A comparative analysis of Facebook, Twitter, Delicious, YouTube, and Flickr, J. Assoc. Inf. Sci. Technol. 66 (2015) 2045–2060.
[23] J. Peng, Y. Zhou, X. Sun, J. Su, R. Ji, Social media based topic modeling for smart campus: A deep topical correlation analysis method, IEEE Access 7 (2019) 7555–7564.
[24] D. Ramachandran, P. Ramasubramanian, Event detection from Twitter - a survey, Int. J. Web Inform. Syst. 14 (3) (2018) 262–280.
[25] S. Stieglitz, M. Mirbabaie, B. Ross, C. Neuberger, Social media analytics - Challenges in topic discovery, data collection, and data preparation, Int. J. Inf. Manag. 39 (2018) 156–168.
[26] I. Tien, A. Musaev, D. Benas, C. Pu, Detection of damage and failure events of critical public infrastructure using social sensor big data, in: Proc. International Conference on Internet of Things and Big Data, Rome, Italy, 2016, pp. 435-440.
[27] A. Troudi, C.A. Zayani, S. Jamoussi, I.A.B. Amor, A new mashup based method for event detection from social media, Inf. Syst. Front. 20 (5) (2018) 981–992.
[28] G. Valkanas, D. Gunopulos, Event detection from social media data, IEEE Data Eng. Bull. 36 (2013) 51–58.
[29] S.F. Waterloo, S.E. Baumgartner, J. Peter, P.M. Valkenburg, Norms of online expressions of emotion: Comparing Facebook, Twitter, Instagram, and WhatsApp, New Media Soc. 20 (2018) 1813–1831.
[30] A. Weiler, M. Grossniklaus, M. H. Scholl, Event identification and tracking in social media streaming data, in: Proc. Workshops of the EDBT/ICDT 2014 Joint Conference, Athens, Greece, 2014 pp. 282-287.
[31] F. Wu, B.A. Huberman, Finding communities in linear time: a physics approach, Eur. Phys. J. B 38 (2004) 331–338.
[32] C. Zhang, S. Lu, C. Zhang, X. Xiao, Q. Wang, G. Chen, A novel hot topic detection framework with integration of image and short text information from Twitter, IEEE Access 7 (2019) 9225–9231.
[33] H. Zhou, H. Yin, H. Zheng, Y. Li, A survey on multi-modal social event detection, Knowl.-Based Syst. 195 (2020), 105695.
[34] X. Zhou, B. Wu, Q. Jin, User role identification based on social behavior and networking analysis for information dissemination, Futur. Gener. Comput. Syst. 96 (2019) 639–648.
[35] R. Esmaeilyfard, M. Naderi, Distributed composition of complex event services in IoT network, J. Supercomput. 77 (6) (2021) 6123–6144.
[36] X. Zhu, Complex event detection for commodity distribution Internet of Things model incorporating radio frequency identification and Wireless Sensor Network, Futur. Gener. Comput. Syst. 125 (2021) 100–111.
[37] X. Chang, Y. Yu, Y. Yang, E.P. Xing, Semantic Pooling for Complex Event Analysis in Untrimmed Videos, IEEE Trans. Pattern Anal. Mach. Intell. 39 (8) (2017) 1617–1632.
[38] X. Chang, Z. Ma, Y. Yang, Z. Zeng, A.G. Hauptmann, Bi-Level Semantic Representation Analysis for Multimedia Event Detection, IEEE Trans. Cybern. 47 (5) (2017) 1180–1197.
[39] E.A. Alomari, I.A. Katib, A. Albeshri, T. Yigitcanlar, R. Mehmood, Iktishaf+: A Big Data Tool with Automatic Labeling for Road Traffic Social Sensing and Event Detection Using Distributed Machine Learning, Sensors 21 (9) (2021) 2993.
[40] E. AlOmari, I. Katib, R. Mehmood, Iktishaf: A Big Data Road-Traffic Event Detection Tool Using Twitter and Spark Machine Learning, Mobile Netw. Applic. (2020) 1–16.
[41] H. Hettiarachchi, M. Adedoyin-Olowe, J. Bhogal, M.M. Gaber, Embed2Detect: temporally clustered embedded words for event detection in social media, Mach. Learn. 111 (1) (2022) 49–87.
[42] Y.M. George, S. Karunasekera, A. Harwood, K.H. Lim, Real-time spatio-temporal event detection on geotagged social media, J. Big Data 8 (1) (2021) 91.