Journal of Intelligent Information Systems (2023) 60:73–95
https://doi.org/10.1007/s10844-022-00730-8

Generalized durative event detection on social media

Yihong Zhang[1] · **Masumi Shirakawa**[1] · **Takahiro Hara**[1]

Received: 16 February 2022 / Revised: 15 July 2022 / Accepted: 18 July 2022 /
Published online: 29 July 2022

Abstract
Given the recent availability of large volumes of social media discussions, finding temporal unusual phenomena, which can be called events, from such data is of great interest. Previous works on social media event detection either assume a specific type of event, or assume certain behavior of observed variables. In this paper, we propose a general method for event detection on social media that makes few assumptions. The main assumption we make is that when an event occurs, affected semantic aspects will behave differently from their usual behavior, for a sustained period. We generalize the representation of time units based on word embeddings of social media text, and propose an algorithm to detect durative events in time series in a general sense. In addition, we also provide an incremental version of the algorithm for the purpose of real-time detection. We test our approaches on synthetic data and two real-world tasks. With the synthetic dataset, we compare the performance of retrospective and incremental versions of the algorithm. In the first real-world task, we use a novel setting to test if our method and baseline methods can exhaustively catch all real-world news in the test period. The evaluation results show that when the event is quite unusual with regard to the base social media discussion, it can be captured more effectively with our method. In the second real-world task, we use the event captured to help improve the accuracy of stock market movement prediction. We show that our event-based approach has a clear advantage compared to other ways of adding social media information.

Keywords Event detection · Social media · Heuristic methods

✉ Yihong Zhang

Masumi Shirakawa
shirakawa@hapicom.jp

Takahiro Hara
hara@ist.osaka-u.ac.jp

[1] Graduate School of Information Science and Technology, Multimedia Data Engineering Lab, Osaka University, Osaka, Japan

1 Introduction

Event detection on social media in recent years has attracted a large number of researches. Given large volumes of social media data and the rich information contained in them, event detection on social media is both beneficial and challenging. With social media text as the base data, important previous works have proposed methods for detecting earthquakes Sakaki et al. (2010), emerging topics for organizations Chen et al. (2013), influenza trends Gao et al. (2018), and public gatherings Khalifa et al. (2017). In these works and many others, however, it is required to have some prior knowledge or assumptions of the potential event. These assumptions include some known keywords or entity names that are associated with the event Sakaki et al. (2010); Popescu and Pennacchiotti (2010); Cataldi et al. (2010); Chen et al. (2013); Unankard et al. (2015); Olteanu et al. (2014), and some manually created labels for events as the supervised training dataset Li et al. (2012); Gao et al. (2018). Furthermore, the definition of event also differs in these works. Some consider an event as a temporal-spatial concentration of similar texts Unankard et al. (2015); Gao et al. (2018); Zhou and Chen (2014); Li et al. (2012); Dong et al. (2015), while others consider it as an unusual burstiness in term frequency Chen et al. (2013); Weng and Lee (2011); Rossi et al. (2018).

In this paper, in contrast, we attempt to provide a general solution to event detection in social media with minimum prior assumption of the event. First of all, we follow a general definition of event that is not restricted to social media data. This definition was proposed by Jaegwon Kim, who considered that an event consists of three parts, a finite set of objects x, a property P, and a time interval t Kim (1976). To better illustrate, let us consider a scenario of an amusement park. Normally, customers wander around the park, visiting different attractions in almost a random manner. When a show starts to perform in the central stage, those who are interested in the show will be moving towards the stage. In this scenario, the object x are the customers who are interested in the show, the property P is the direction of the stage, and the time interval t is the duration of the show. Note that just as not all customers in the park are interested in the show, $x \in X$ in an event is a subset of all possible objects.

Putting it on the social media case, when an event creates an impact on people's lives, it is likely that it will be reflected on online discussions. Certain semantic aspects of posted texts, which can be considered the object set x, would suddenly have unusual trends together, whose deviation can be considered the property P, for the duration of the event t. For example, around June every year, certain words would appear around the topic of "summer", including "summer holiday", "ice cream", "air conditioner" and "firework festival". As another example, when there is a missile launched by a neighboring country, for a moment we can see words related to this event, including "missile launch", "evacuation", "disaster-prevention kits", and some location names. Such examples illustrate that an event can invoke unusual activities of certain semantic aspects in social media discussions. While individual words or semantic dimensions may be vague (e.g., "festival"), the combination of them can show richer meanings (e.g. "firework festival in summer"). The problem then is how to capture x, P and t in social media text through a computational method.

The principle of our design is to make as few assumptions about the event as possible. Here are two assumptions we make in our method. First, there is a finite set of components in the system, and a subset of components will be affected by the event. Second, for the duration of the event, affected components behave differently

from their usual, normal behavior. We consider these are minimum assumptions that are within restrictions in Kim's definition of an event. Given these assumptions, our method takes two steps to achieve event detection. First, we convert unstructured social media text data into distributed representation, also called *word embeddings*, where each dimension represents a semantic aspect, and is considered a component in the system. This can be done with existing distributed representation learning techniques, also called embeddings. In this paper we use word2vec Mikolov et al. (2013) instead of other embedding methods such as GloVe Pennington et al. (2014), due to its availability and proven effectiveness for different languages, especially Japanese. Note that in this paper we consider only social media text. However, the images in social media can be studied in a similar way as they are turned into multi-dimension vector representations using models such as Inception Szegedy et al. (2016). Second, we design and use a multi-dimension anomaly detection algorithm to capture the unusual behavior, with a customizable normality test. The algorithm detects abnormal intervals in single time series and combines them to form affected components of an event by finding the intersections. This design can be considered *bottom-up*, which generates a high-level event from known low-level events, and the high-level event contains a more distinct meaning. For example, an event "firework festive in summer" is generated from "firework", "festival", and "summer" events. We note here, though, that the problem could be approached the other way around, i.e, a *top-down* approach. Such an approach may need to first detect the holistic event, and then detect abnormality in sub-dimensions, which cannot be detected individually. It will require a very different set of methods, and thus will not be addressed in this paper.

Our method is general in two ways. First, our method generalizes social media text into semantic aspects. With this generalization, we now look at the collective behavior of social media posts instead of tracking individual term frequency. This is useful in many scenarios. For example, during New Year holiday in Japan, many aspects of the real-world phenomenon become visible, including New Year's meal (

年越し

), a specific TV program (

紅白

), New Year's greeting (

挨拶

), and the general happy mood. Individually, these terms may not have a significant frequency change, but collectively, they make the New Year event unusual. Second, our method generalizes event detection as anomaly detection in time series. In contrast to previous works, we deal with durative events instead of punctual events. With a customizable normality test function, we can detect events with arbitrary lengths. Such generality allows our method to be applicable to a wider range of tasks than previous works. Since our method is straightforward to implement, future extension can be easily made for the need of specific tasks.

In a previous conference presentation, we showed the algorithm for retrospective event detection that considered only past data, and discussed the experimental results in the

real-world task of newsworthy words recommendation[1] Zhang et al. (2021). Some materials from the presentation are used in this paper. On top of it, in this paper, we propose an incremental version of the algorithm which can be potentially used in real-time event detection. By reducing the unit of analysis from the frequencies of thousands of words to one or two hundreds of embeddings, and offering real-time capability, our method can be potentially effective for coping with the fast change rates of today's big data. Furthermore, we introduce a set of experiments on a synthetic dataset to test the new method, and a new evaluation task of stock price movement prediction with real-world data. We organize the remainder of this paper as the following. In Section 2, we will discuss related works on event detection in social media. In Sections 3 and 4, we will present our method to generalize social media text to temporal word embeddings, and to detect unusual behavior in them in a retrospective way. In Section 5, we will present the incremental version of the algorithm. In Section 6, we will present and show the results of experiments using a synthetic dataset. In Section 7, we will present experimental evaluation of our method in a real-world evaluation task of recommending newsworthy words. In Section 8, we will test our approach in another real-world task of stock market movement prediction. Finally Section 9 will conclude this paper.

2 Related work

Previous surveys on social media event detection works have commonly divided works according to whether detected events are specific or non-specific Atefeh and Khreich (2015); Saeed et al. (2019). Here we would like to provide a new aspect of events in existing works, that is whether events are considered one-time events or events lasting for a period of multiple time units. In other words punctual and durative events.

2.1 Punctual event detection

Essentially, punctual events are supposed to be the point of drastic change in the observed variables Guralnik and Srivastava (1999). While this limits the phenomenon they can represent, events with this definition are indeed easier to capture, and many works followed this approach. For example, the Twitter-based earthquake detection system proposed by Sasaki et al. Sakaki et al. (2013) raises an alarm at the moment when the number of tweets classified as earthquake reports reaches a certain threshold. Similarly, the event detection system proposed by Zhang et al. raises an alarm at the moment when the number of incident reports within a geographical region reaches a threshold Zhang et al. (2018). Zhao et al. proposed a probabilistic model for detecting spatiotemporal events at the time they happen Zhao et al. (2016). Weng and Lee proposed an event detection method based on wavelet transformation and word clustering Weng and Lee (2011). An event flag is set for a time slot if frequency correlation of co-occurring words is larger than a threshold. The crime and disaster event detection system proposed by Li et al. aims to extract the time an event happened, by location estimation and geographical clustering Li et al. (2012). The location-based event detection method by Unankard et al. also uses a threshold to decide if an event has happened, by comparing the frequency in the current and previous time unit

[1] The preprint can be accessed online at http://arxiv.org/abs/2106.02250

Unankard et al. (2015). The disaster monitor system by Rossi et al. decides if an event happened by determining if word frequency in the current time slot is an outlier Rossi et al. (2018). Xie et al. proposed another one-time event detection method for Twitter, using a two-stage monitoring scheme Xie et al. (2016). Following this work, Zhang et al. proposed a refined method that focused on detected topic coherence Zhang et al. (2017). Hua et al. proposed an event detection system for Twitter that classifies tweets into topics based on labels generated from news Hua et al. (2013). Their method, however, does not consider the time factor. Khodabakhsh et al. proposed a method to predict personal life-event by tracking user past tweets Khodabakhsh et al. (2020). Their method, however, has limited applicability because it relies on pre-define categories of life events. There are also some works that proposed to detect events retrospectively, but with event dates and other information identified from text. For example, Ritter et al. proposed such a method to extract open events from tweets, built upon natural language processing techniques Ritter et al. (2012). Parikh and Karlapalem proposed a system to extract information from temporally coherent tweets by clustering the keywords Parikh and Karlapalem (2013). Suliman et al. proposed to extract information from events detected based on word frequency Suliman et al. (2016). While information extraction seems to be an important task in these works, it is not a focus in this paper. Nevertheless, in our experimental evaluation, we offer a method to extract verbal information from detected events.

2.2 Durative event detection

While not uncommon in time series pattern mining Batal et al. (2012); Minnen et al. (2007); Vahdatpour et al. (2009), compared to punctual events, social media event detection methods that follow a durative event definition are rather scarce. Relevant works include the emerging topic detection method proposed by Chen et al., which identifies two time points, the moment the topic starts and the moment the topic becomes hot Chen et al. (2013). The purpose of the method is to identify emerging topic before the topic becomes hot, and detected events thus last for periods of varied lengths. One requirement of the method, however, is that the tweets collected should be related to certain organizations, which makes the method less applicable. Walther and Kaisser proposed a geo-spatial event detection method that can be considered durative because the input data has fixed periods Walther and Kaisser (2013). However, constraining dataset period to achieve the effect of durative analysis is expensive and difficult to scale. The multiscale event detection method proposed by Dong et al. Dong et al. (2015) aims at discovering events with spatio-temporally concentrated tweets. Without a preset time length for the event, the method clusters tweets that have similar spatio-temporal context, and thus indirectly detects events that last for a period. However, the requirement of spatial information also limits the applicability of the method. In this paper, on the other hand, we aim at providing a general method for detecting durative events with less restrictions. Another interesting work proposed by Kleinberg attempted to use states to detect durative events Kleinberg (2003). His method can be useful when the emerging words are at different stages of emergence. However, this method was applied only to single words. In contrast, our work tries to detect structural events composed of multi-dimension information. Nevertheless, in some social media analysis tasks, such as the one described in Section 7, we can compare the effectiveness of his method and ours using the same metric, even though the underlying mechanisms are different.

3 Generalized representation of temporal social media text

We first deal with the problem of representing temporal social media text in a general way. A simple way to represent social media text is through bag-of-words (BOW). BOW representation essentially considers that words in text are independent tokens, and each document is a collection of them. There are two problems with BOW representation. First, in a large text collection, the vocabulary is also large, usually includes thousands of words, and tracking temporal activity of each word is computationally expensive. Second, considering words as independent tokens ignores semantic information about words, which may be important for event detection. For example, Covid-19 and Corona are both names of the virus in the current pandemic, and should be considered together in one event, but BOW representation would consider them separately.

To mitigate these problems, we propose to use *word embeddings* to represent temporal social media texts. First proposed by Mikolov et al., word embeddings are distributed representation of words learned from text contexts Mikolov et al. (2013). The learning technique extracts the surrounding words of a certain word and encodes them in a neural network encoder, so that a vector, called an embedding, can be associated with the word, and each element in the vector represents a certain semantic aspect of the word. While the meaning of the semantic aspect of the embedding is difficult to be understood by human reader, it has been shown that words with similar embeddings would have a similar semantic meaning. For example, *apple* would have a more similar embedding to *orange* than to *bird*.

Using word embeddings thus mitigates the problems of BOW representation. First, it reduces dimensionality. Typical word embeddings would have between 50 and 300 dimensions. Second, it allows consideration of semantics, so that words of similar meanings can be considered together. By considering semantics, we actually generalize text into a more abstract level. For example, when detecting the pandemic event, we no longer deal with individual words such as Covid-19 and Corona, but the virus or disease these words refer to. Given it is effectiveness, previous works have already used word embedding to represent not only text documents, but also users and spatial units such as locations Wang et al. (2018); Shoji et al. (2018). In this work, we utilize word embeddings to generate vector representations of time units.

To generate vector representation for a time unit, we take the following steps.

1. assigning collected text messages to time units.
2. tokenizing text messages so that words are also assigned to time units
3. obtaining word embeddings for assigned words
4. the vector representation for a time unit is taken as the average value of all embeddings of the words assigned to the time unit

We can use existing natural language processing libraries to segment and turn tweets into words. To obtain word embeddings, we can use existing implementations of *word2vec* and a general purpose training corpus such as Wikipedia[2]. Word embedding learned under such setting would represent words with their general meaning in daily usages. The final result of this process is a vector representing the totality of social media discussions for each time unit.

[2] An example online resource that provides an implementation under this setting: https://github.com/philipperemy/japanese-words-to-vectors

4 Generalized multi-dimension event detection in time series

At this point we have a vector for each time unit representing social media discussions. The next task is to detect events from such representations. In a way this representation can be seen as multivariate time series data, with each dimension as one observed variable. While there are previous works that have proposed event detection for time series data, most of them are dealing with punctual event Guralnik and Srivastava (1999); Cheng et al. (2009), or require the events to be repeating and predictable Batal et al. (2012). In this work, we accept the hypothesis that an event is something that cannot be predicted, thus the behavior of affected components cannot be pre-defined Taylor and Williams (2009). We aim to make minimum assumptions about the event, and the main assumption we make is that when affected by an event, the component will behave differently from its usual behavior.

Our method detects multi-dimension event from multivariate time series in two steps. First it detects unusual intervals of observations in a single dimension (Algorithm 1). Then given a list of abnormal intervals in each dimension, it finds basically the intersections of abnormal intervals, and outputs them as multi-dimension events (Algorithm 2).

Algorithm 1 Find largest intervals with significant alternation to normality

INPUT: TS, k_{min}, k_{max}, f_n, δ
OUTPUT: a list of intervals Is

1: $Is \leftarrow \{\}$
2: $i \leftarrow 1$
3: **while** $i < (|TS| - k_{min})$ **do**
4: $\quad largest\_interval \leftarrow \{\}$
5: $\quad$ **for** j in $(i + k_{min})$ to $min(|TS|, i + k_{max})$ **do**
6: $\quad\quad$ **if** $f_n(TS \setminus TS(i,j)) - f_n(TS) > \delta$ **then**
7: $\quad\quad\quad largest\_interval \leftarrow (i,j)$
8: $\quad\quad$ **end if**
9: $\quad$ **end for**
10: $\quad$ **if** $largest\_interval$ is empty **then**
11: $\quad\quad i \leftarrow i + 1$
12: $\quad$ **else**
13: $\quad\quad Is \leftarrow Is \cup largest\_interval$
14: $\quad\quad i \leftarrow (b \text{ in } largest\_interval) + 1$
15: $\quad$ **end if**
16: **end while**

Shown in Algorithm 1, we design an algorithm to find the largest interval with significant alternation to normality. It takes a univariate time series as input, as well as two parameters k_{min} and k_{max}, which are the minimum and maximum number of time units for the detected intervals. It also requires a customizable function f_n for the normality test, and a corresponding threshold δ. The algorithm starts from the beginning of the time series (line 2, 3). At each time point i, it tests all intervals that ends between $i + k_{min}$ and $i + k_{max}$ (line 5). With each interval, it performs normality test with the specified function f_n, and if the normality difference between the time series with and without the interval is larger than δ, then the interval is considered abnormal (line 6). The largest interval considered abnormal will be taken as the abnormal interval starts at time i (line 7). If an abnormal interval is found, the algorithm will move to the end of the interval (line 13, 14), and continue until it reaches the end of the time series. Finally the algorithm returns all abnormal intervals found as Is.

Algorithm 2 Find multi-dimension events

INPUT: Is, k_{min}, c_{min}
OUTPUT: E
$E \leftarrow \{\}$
$E_{half} \leftarrow \{\}$
for i in 1 to $n - k_{min}$ **do**
  $D_{cur} \leftarrow \{d_j | i \in Is_j\}$
  $D_{old} \leftarrow \{\}$
  for each $e_{half} \in E_{half}$ **do**
    $D_{continuing} \leftarrow d(e_{half}) \cap D_{cur}$
    if $D_{continuing} = \{\}$ **then**
      next
    end if
    remove e_{half} from E_{half}
    if $|D_{continuing}| > c_{min}$ **then**
      $e_{continuing} \leftarrow (start(e_{half}), i, D_{continuing})$
      $E_{half} \leftarrow E_{half} \cup e_{continuing}$
      $D_{old} \leftarrow D_{old} \cup D_{continuing}$
    else
      $D_{continuing} \leftarrow \{\}$
    end if
    $e_{finished} \leftarrow (start(e_{half}), i, d(e_{half}) \setminus D_{continuing})$
    if $l(e_{finished}) > k_{min}$ & $|d(e_{finished})| > c_{min}$ **then**
      $E \leftarrow E \cup e_{finished}$
      $D_{old} \leftarrow D_{old} \cup (d(e_{finished}) \cap D_{cur})$
    end if
  end for
  $D_{new} \leftarrow D_{cur} \setminus D_{old}$
  if $|D_{new}| > c_{min}$ **then**
    $E_{half} \leftarrow E_{half} \cup (i, i, D_{new})$
  end if
end for

It is worth noting that Algorithm 1 does not necessarily find intervals that deviate most from normality. For example, given a highly abnormal interval *I*, a few time units surrounding *I* may be normal by themselves, but when considered together with *I*, this larger interval may still be abnormal above the threshold. And our algorithm will pick the larger interval instead of the more deviating interval. Since our goal is to detect multi-dimension events, and the intervals are to be taken as the input of next step, it is rather desirable to have the largest possible abnormal intervals, instead of smaller, more deviating intervals.

The normality test function f_n can be defined by the user, as long as it outputs a score for data normality or randomness. There are many existing normality test functions available to use, including Box test and Shapiro Wilk test Bartels (1982). For the completion of the method, we use the rank version of von Neumannâ€™s ratio test Bartels (1982) in our experimental analysis[3]. After trying a few test functions, we found that this randomness test captures unusual intervals in data more consistently.

After processing the data with Algorithm 1, we now have a list of abnormal intervals *Is* for each of the word embedding dimension. The goal of next algorithm, shown as Algorithm 2, is to find the intersection of these intervals. It is an incremental algorithm that

[3] An implementation of this test is available as an R package: https://cran.r-project.org/web/packages/randtests/randtests.pdf

needs to go through the dataset only once. It takes the set of *Is* as inputs, as well as two parameters, k_{min} as the minimum length of an event period, and c_{min} as the minimum number of affected dimensions in an event.

At each time point i, the first thing to do is find the dimensions that behave unusually at i, based on the intervals detected (line 4). From there, these dimensions are either considered a part of a continuing event, or put to form a new event. We always keep a list of events that are halfway through E_half, and at each time point, we check through all halfway events for continuity (line 2, 6). If affected dimensions at time i match halfway events, they are assigned to these events, and if enough dimensions are assigned ($> c_{min}$), the halfway event is considered continuing (line 7 to 18). If a halfway event could not be matched with enough affected dimensions, the event is considered finished (line 19 to 23). Those dimensions not matched with any halfway event are grouped to form a new halfway event, if there are enough of them (line 25 to 28). The final output is a list of events E, where each $e \in E$ has $e = \{\mathbf{x}, t\}$, with $\mathbf{x}$ as affected dimensions, and t as the event period.

5 Incremental multi-dimension event detection in time series

In the previous section we show an algorithm to detect multi-dimension events by comparing the normality of segments with the entire time series. This algorithm works if we are allowed to find events by retrospectively examining past data. However, in many situations, we are interested in detecting current on-going events, and such events cannot be detected retrospectively since data surrounding the event is not available. What we usually have, instead, is a segment of current data, in addition to some past data. In this section, we present an algorithm to incrementally detect multi-dimension events with such data.

Assuming we have some past data, which are time series of all dimensions. We call this data *TSs_REF*, as we will use it as the reference to compare normality. Then at each time segment t, we obtain the time series data within the current segment, call *TSs_CUR*. The idea is to compare normality of the data with and without *TSs_CUR* on top of *TSs_REF*. The algorithm is shown in Algorithm 3.

Algorithm 3 Find events from time series at time t

INPUT: TSs_REF, TSs_CUR, k_{min}, c_{min}, f_n, δ, E_{half}
OUTPUT: E, E'_{half}

1: $E \leftarrow \{\}$
2: $D_{cur} = \{\}$
3: **for** j in 1 to $|TSs|$ **do**
4: **if** $f_n(TSs\_REF_j) - f_n(TSs\_REF_j \cup TSs\_CUR_j) > \delta$ **then**
5: $D_{cur} \leftarrow D_{cur} \cup j$
6: **end if**
7: **end for**
8: follow line 5-28 in Algorithm 2
9: $E'_{half} \leftarrow E_{half}$

The basic setting of the algorithm is similar to Algorithms 1 and 2. We need to define k_{min}, c_{min}, and the normality function f_n and threshold δ. We also assume some unfinished events are carried on from the previous time segment, denoted as E_{half}. Then the algorithm is different from the retrospective algorithm in that it does not detect affected dimensions by first detecting abnormal portions. Instead, it detects affected dimensions on-the-fly by comparing

normality test on the reference data, with and without the addition of current time segment data (line 4-5). After detecting the affected dimensions, we can follow exactly line 5-28 in Algorithm 2 to detect the events. In other words, the abnormal dimension detected in the current time segment will either be added to an ongoing event or start a new event with potentially other abnormal dimensions. At the end of the algorithm, it outputs not only completed events E, but also new unfinished events E'_{half} to be considered in the next time segment.

6 Evaluation on synthetic dataset

In the first set of experiments, we test our approach on a synthetic dataset. Assuming social media data has been transformed into time series. The synthetic data is generated as time series that is affected by a number of multi-dimension events. In this section, we will discuss the synthetic data generation process and the evaluation of proposed methods on this dataset. The event detection algorithms and synthetic data generation program are implemented using the R language[4]. The code is available upon request. The experiments, as well as experiments in the following sections, have been performed on a Windows PC with an Intel Core i7-8700 CPU and 64 GB memory.

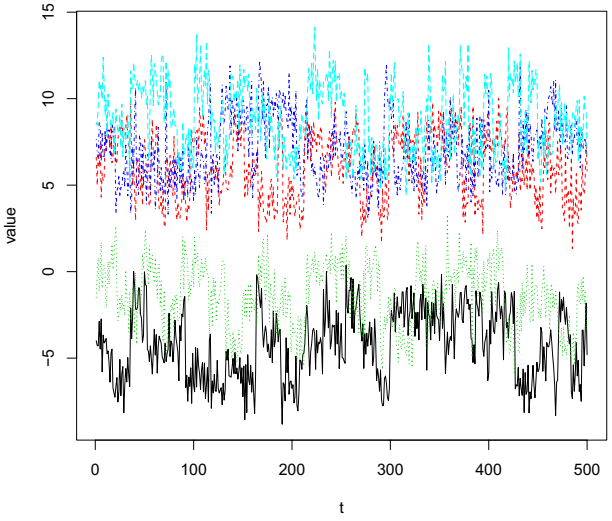
6.1 Generating a synthetic multi-dimension event dataset

We generate a synthetic multi-dimension event dataset in two steps. In the first step we generate normal data. Given n time steps and m dimensions, we generate m time series of length n. For each time series (i.e., each dimension), the values are obtained by sampling a normal distribution of a mean μ_d and a sd of 1. The mean μ_d is an integer randomly chosen from $[-10, 10]$. In the second step, we add multi-dimension events into the dataset. With a randomly chosen event length l, begin time b, affected dimensions D, where each dimension is associated with a property p_d randomly chosen either as -1 or 1, we create o events. For affected dimension $d \in D$ in each event, we alter the normal data by replacing values between $b, b + l$ with event values. The event values are drawn from a random distribution with a mean of $\mu_d + 2 * p_d$. In this way, when a dimension is affected by an event, it will deviate from the normal behavior either positively or negatively, depending on the value of p_d. In our experiment, we set $n = 500$, $m = 100$, $o = 1000$, minimum and maximum event lengths as 5 and 20, minimum and maximum numbers of affect dimensions in an event as 3 and 7, and generate the synthetic dataset. Parts of the dataset consisting of 5 dimensions are shown in Fig. 1. As a comparison, we also show some parts of real-world social media data represented by word embedding time series. As we can see, although the synthetic data generally exhibits a larger variance, it is very close to real-world data in terms of tendencies.

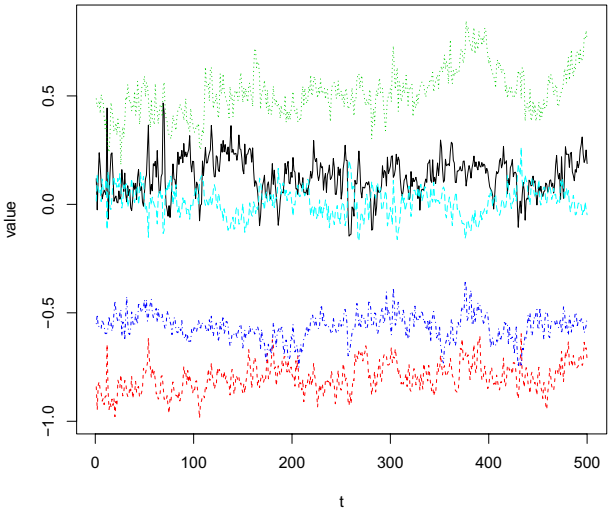
6.2 Evaluation metric

When generating the synthetic data, we also recorded the information for generated events, including beginning time, ending time, and affected dimensions. Evaluating our event detection method requires comparing the detected events with such ground-truth

[4] https://www.r-project.org/



(a)



(b)

Fig. 1 Parts of synthetic data in comparison with real data

even information. Since there is randomness involved in the generation process, it is hard to expect that the detected events will match exactly the event in ground truth. Therefore we allow some flexibility when matching the events. Specifically, we use two tolerance thresholds tol_t and tol_d. We consider a detected event $e_d \in E_d$ matches a ground truth event $e_g \in E_g$ if the differences in starting time and ending time are less than tol_t, and the affected dimensions in two events differ by no more than tol_d.

Based on the event matching, we calculate two performance metrics. First we calculate *recall*. For each $e_g \in E_g$, we compare it with all the detected events $e_d \in E_d$. If a match is found, we add one to the true positive counter *TP*, and move to the next event

Table 1 Recall of two methods with different tolerance

(tol_t,tol_d)	(3,2)	(3,3)	(5,2)	(5,3)	(8,2)	(8,3)
whole	0.026	0.132	0.046	0.222	0.084	0.272
incremental	0.024	0.142	0.049	0.215	0.077	0.275

in E_g. It is guaranteed $TP < |E_g$. The recall is then calculated as $\frac{TP}{|E_g|}$. Then we calculate *precision*. For each $e_d \in E_d$, we compare it with each $e_g in E_g$. If a match is found we move to the next event in E_d, such that it is guaranteed $TP < |E_d|$. The precision is then calculated as $\frac{TP}{|E_d|}$. We use two separate processes to calculate recall and precision because it is possible that more than one event in a set can match an event in the other set, given the flexible matching method.

6.3 Evaluation results

We show the results of the whole and incremental event detection methods with different tolerance values in Table 1 and 2. We test three tol_t values including 3, 5, and 8, and two tol_d values including 2, 3. First we look at the performance of the *whole* method. If the tolerance is strict, such as allowing 3 time difference and 2 dimension difference, only 2.5% in the ground truth event can be detected, and only 11% detected events finds a match event in the ground truth. On the other hand, when tolerance is relaxed, such as allowing 8 time difference and 3 dimension difference, the method can detect 27% ground truth events, while 90% detected events finds a matching event in ground truth. Next we compare the performance of whole and incremental method. We find that even though the incremental method only considers data in current time segment, its performance is almost the same as the whole method, especially when considering the recall. For precision, the incremental method is slightly worse than the whole method.

7 Application scenario: Recommending newsworthy words

Since the result of event detection using our method is difficult to interpret by human reader, we use real-world social media data to verify the effectiveness of our method indirectly. We extend our method to perform a task called recommending newsworthy words, which has been the evaluation task in other event detection works Unankard et al. (2015); Dong et al. (2015). We will present the details of this task and the results

Table 2 Precision of two methods with different tolerance

(tol_t,tol_d)	(3,2)	(3,3)	(5,2)	(5,3)	(8,2)	(8,3)
whole	0.117	0.582	0.188	0.808	0.347	0.906
incremental	0.077	0.495	0.158	0.752	0.267	0.923

in this section. It is worth noting here, though, that our event detection can potentially do more than recommending newsworthy words.

7.1 Evaluation task

Given a set of time units $T = \{t_1, ..., t_c\}$, for each time unit, we apply the event detection method on a social media discussion dataset, and generate a ranked list of event words P from detected events. Also for each time unit, we generate from news sources a ranked list of news words G. The evaluation is done by comparing P and G. If $|G \cap P|$ is large, then the event detection method is considered capable of capturing newsworthy words, which also shows that the news has an impact on the social media discussion.

Traditional evaluation of event detection is centered on detected events Unankard et al. (2015). It verifies whether detected events is corresponding to real-world events, and does not do anything when a real-world event has not been detected (false negative). We on the other hand, attempt an exhaustive evaluation that concerns all real-world events happened. Specifically, we consider all news headlines from news source for each time unit, and evaluate to what degree corresponding information can be detected by the event detection method.

7.2 Social media discussion dataset

Since it is not feasible to monitor all messages in a social media platform such as Twitter, we select a subset of all messages on Twitter as our social media discussion dataset. First we obtain a list of Japanese politician Twitter accounts[5]. Then we monitor all tweets mentioning these accounts using Twitter Stream API[6]. For a period of six months between January and July, 2020, we collected about 6.9 million tweets, after removing retweets. We take this as the discussion dataset. We understand this dataset does not represent the overall discussion happening on Twitter, but rather has a focused theme that is Japanese politics. But such discussions and the community producing them may still be affected by general news, and it will be interesting to see what unusual events can be captured from these discussions and how they correspond to news sources. It is expected that if we can detect the events in this discussion dataset, we can also detect events in the discussion of different themes in the same way.

We store the raw text from tweets and perform basic cleaning, including replacing change-line characters with white spaces, removing unrecognizable Unicode characters, and removing the url links in the text. Then we use the natural language processing package kuromoji[7] to further process the Japanese text. The package can effectively perform segmentation and part-of-speech (POS) tagging for Japanese text. After POS tagging, we select only nouns to represent the information in the text. We also filter out some less frequent words, and consider only 8,267 words that have appeared at least 500 times in the dataset. The key information about the dataset is shown in Table 3.

[5] Since politician are public, such a list can be found in many online sources, for example: https://meyou.jp/group/category/politician/

[6] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data

[7] https://github.com/atilika/kuromoji

7.3 Ground truth generation

We generate ground truth news words as follows. First we collect messages posted by a number of Japanese news Twitter accounts[8]. Among 916 news accounts considered, some are general news accounts reporting local and international news, some are specific news accounts reporting news for example in sports or entertainment. Messages sent from these accounts are usually news headlines. To make our target clearer, we select from collected messages three specific topics, namely, *politics*, *international*, and *Corona*. The selection is done by filtering collected messages with these three topic words as hashtags. During a one-month period between June and July, 2020, we collected 814 political news headlines, 503 international news headlines, and 602 Corona news headlines. These news headlines are assigned to time units of one hour length.

We turn these news headlines into nouns by the same kuromoji software described in the previous section, and count the frequencies. These words are then ranked using *tfidf*, which is calculated as:

$$tfidf(w) = tf(w) \cdot \log \frac{|D|}{|d \in D : w \in d|}$$

where $tf(w)$ is the frequency of word w, and D is a collection of documents, which in our case is messages assigned to $|D|$ time units. Finally, for each time unit, we pick top-20 words ranked by *tfidf* as the ground truth news words.

7.4 Recommending newsworthy words from detected events

Since our method does not generate ranked words directly, we need a method to convert the output of our method into words. The output of our method is a list of events $E = \{e_1, ..., e_m\}$, where for each event we have a set of affected dimensions $\mathbf{x}$ and duration t.

To convert this result back to words, we first calculate the deviation of an affected dimension in the event duration as the difference between mean value of the dimension in the event duration, and the mean value outside the duration:

$$dev_e(x) = mean\_freq(x, t) - mean\_freq(x, \neg t)$$

which can be considered a part of event property P. Then for each word w with embedding $embedding_w$, an event score is calculated as the product of the embedding value and the deviation in the affected dimensions:

$$event\_score_e(w) = \sum_{x \in \mathbf{x}} embedding_w(x) \times dev_e(x)$$

In this way, words with the same deviation tendency as the affected dimensions will have a higher score. Finally, to calculate a word score in a time unit, we have

$$time\_score(w) = \sum_{e=1}^{m} event\_score_e(w)$$

[8] A list of popular Japanese news Twitter accounts can be found on the same source: https://meyou.jp/ranking/follower_media

Table 3 Key information of the political discussion dataset

number of tweets	6,932,698
number of users	716,208
size of dictionary	8,267
minimum word frequency	500
maximum word frequency	759,590

which gives higher scores to words with higher event scores in multiple events. The time score is thus used to rank the words in each time unit.

7.5 Baseline methods

We compare our method with three baseline methods in this evaluation task. The first is a *tfidf*-based method commonly used in previous works. In the same way we generate ground truth, we apply the method to the social media discussion dataset and obtain a *tfidf* score for each word in each time unit. Essentially, with this method, we make a comparison of *tfidf*-ranked words between base source, which are social media discussion tweets, and the reference source, which are news tweets.

The second baseline method is based on the Shannon's Wavelet Entropy (SWE). This method is proposed in a Twitter event detection work by Weng and Lee Weng and Lee (2011), and can be adopted for news word recommendation. From the *tfidf* time series of each word in the social media discussion dataset, the method first performs a wavelet transformation to learn a wavelet function ψ and a coefficient C. The coefficient C can be interpreted as the local residual errors. Then an energy value E is calculated as

$$E = \sum_{k} |C(k)|^2$$

where k indicates k-th coefficient. Then the Shannon's Wavelet Entropy is calculated as

$$SWE = -\sum_{j} \rho_j \cdot \log \rho_j$$

where $\rho_j = E_j / E_{total}$, j indicates the j-th time unit in the time slide. SWE measures how unpredictable of the time series in a time slide t, and it will be a higher value when residual errors are more even in the time slide. Once the SWE is obtained, a score can be assigned to a word for ranking.

$$s(w) = \begin{cases} \frac{SWE_t - SWE_{t-1}}{SWE_{t-1}}, & \text{if } SWE_t > SWE_{t-1} \\ 0, & \text{otherwise} \end{cases}$$

which means if SWE of a word is increasing, it will get a higher score. In our experiments, we use the R package *wavethresh*[9] to perform the wavelet transformation and obtain coefficient C.

The third method is a burst enumeration method based on word frequencies (burst enum.). It was proposed by Kleinberg as a detector of word burst strength Kleinberg

[9] https://cran.r-project.org/web/packages/wavethresh/wavethresh.pdf

(2003). It applies a multi-state automaton to describe the state of burst, given a time series of word frequencies. Here we use the enumerate version proposed in the paper, as it can lead to ranking of words. It describes words as having two states, bursty and non-bursty. In order to move from non-bursty to bursty, the frequency change needs to surpass a penalty. More specifically, the cost of a state is defined as:

$$\sigma(i, r, d) = -\ln\left[\binom{d}{r} p_i^r (1 - p_i)^{d-r}\right]$$

where r is the frequency of the word, d is the total number of words, $i \in (0, 1)$ is the two states, i.e., bursty and non-bursty states. The transient probability p_i is predefined to describe how likely the word will change into a state. Since moving from non-bursty state to bursty state should be more difficult, a penalty $\gamma \ln n$ is added to the cost for this transition. The transition from bursty state to non-bursty state does not have this penalty. The next state of the word is thus determined by which state has the lower cost. And then, if a word is in the bursty state, the strength of the burst can be enumerated as:

$$\sum_{t=t_1}^{t_2} (\sigma(0, r_t, d_t) - \sigma(1, r_t, d_t))$$

where t_1 and t_2 is the beginning and the ending time of current burst. In other words, this enumeration shows the improvement in cost incurred by the bursty state over the interval rather than non-bursty state. Words of larger burst enumeration can be considered having more prominent elevated activity.

7.6 Evaluation results

Evaluation results measured as Recall@K and Precision@K are shown in Fig. 2, where K is the number of recommended words. We compare our methods (event whole and event incremental) with *tfidf*, SWE, and burst enumeration methods. A number of Ks are taken between 20 and 200. The higher the result in recall@K means the more words in ground truth are recommended by the method. The higher the result in precision@K means the more words in the recommended words are really newswothy words.

At the first glance, we can see that generally, *tfidf* performs better for the political news, while event methods perform better for the Corona news. The incremental event method achieves almost the same performance as the whole event method. SWE method performs better for the international news, although only slightly better than the event method when k is small. Burst enumeration method follows a similar trend as the event method. It performs worse than *tfidf* for the political news, but better for international news and Corona news. Both being real-time durative event detection methods, the incremental event detection method improved the recall and precision by up to 13%, compared to burst enumeration, for the Corona news, and up to 60% for the international news.

Next we show some examples of recommended newsworthy words by different methods. We pick a day (07/18/2020) and show the top 10 recommended newsworthy words in Table 4. We can see that *tfidf* tends to pick up politics-related words, while burst enumeration and event methods give more weights to the going pandemic. Additionally, burst enumeration picks up travel-related words, while event method picks up education and sports events.

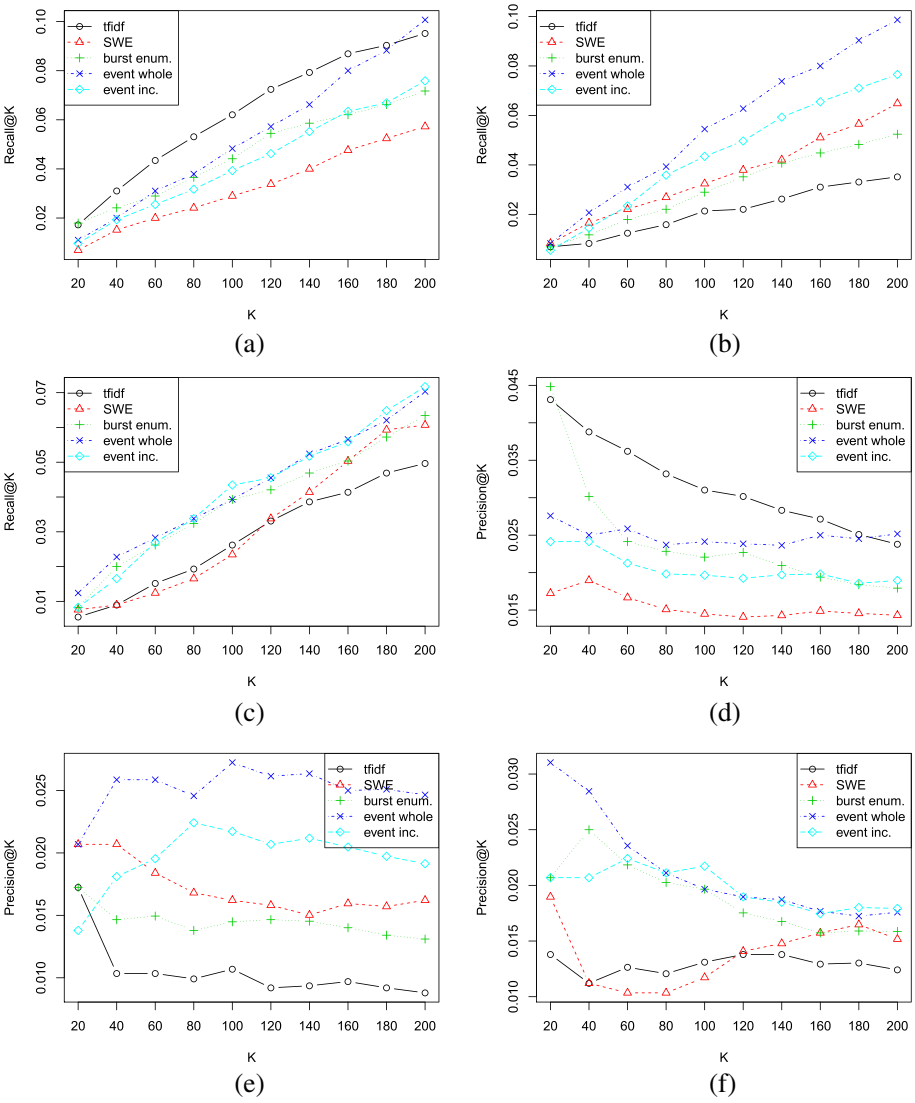


Fig. 2 Recall@K and Precision@K results for three news categories

We now attempt to explain the differences in the results. First thing to note is that recommended words from a method is the same for all three news categories. Since words from news categories are quite different, with limited space, a method better at recommending words for one news category will be worse for other categories. And we can see the results are showing different strengths and weaknesses from different methods. The reason comes from different interpretations of what is news by different methods. For the *tfidf* method, news is considered rises of word usages, and thus words closer to the theme of the social media discussion will be more likely to be recommended. For the event method, news is

Table 4 Top 10 recommended newsworthy words on 07/18/2020

tfidf	Tomita (politician), taiyo, swimming, campaign, middle way, this world, help, Oki-no-tori Island, public election, combined force
SWE	personal, part-time job, moving out, constitution, relaxing, graduation, fund, these people, good judgement, family
burst enum.	tourism, bad policy, travel, isolation, prefectures, school closing down, industry, test negative, Kanagawa, relatives
event inc.	studying, holding (events), supervision, fever, attending school, wearing (mask), business closing down, sports player, holiday, swimming

considered something quite different from the usual state of the discussion, and thus words different from the social media discussion will be more likely to be recommended. And indeed we understand that, since the social media discussion is generally related to politics, political news is more similar to the discussion, while Corona news is more different from the discussion. That is why we see *tfidf* performing better for political news, and event method performing better for Corona news. The burst enumeration method achieves similar results as the event method due to the similar focus on abnormality across time, but it is less sensitive than the event method thus shows an inferior prediction accuracy.

8 Application scenario: stock market movement prediction

Although the event discovered with our method are difficult to interpret for human understanding, it can be easily used in various computational tasks. Particularly, it can be easily transformed into what is so-called social media background Zhang et al. (2021) that can be used in various temporal computational tasks. We will demonstrate the utility through one popular task, that is to predict stock market movement Qian and Rasheed (2007). In this section, we will first describe the task, before showing the experiment settings and describing the evaluation results. Although this is a small-scale experimentation, it is sufficient to show the effectiveness of our approach.

8.1 Evaluation task

Due to its obvious benefits, stock market movement prediction using computational methods has attracted a large number of researches Bollen et al. (2011). Several previous works have considered social media information and other types of data as contextual information that may help improve prediction accuracy Sul et al. (2017). In line with existing works, we consider the task as a binary classification problem Xu and Cohen (2018). More specifically, the problem is to predict whether the price at time p_t is raising or falling compared to the price at time p_{t-1}, given a number of historical prices $p_{t-k}, ..., p_{t-1}$.

The supervised machine learning approach deals with this task by creating a supervised training dataset. Given n historical prices, a training instance is created with $x = \{p_{i-k}, ..., p_{i-1}\}$ and $y = 1$ if $p_i > p_{i-1}$ and 0 otherwise. We can also add contextual information to x. The contextual information can be any vectorized temporal data aligned to the same time. Once we have a supervised training dataset (x, y), we can apply supervised machine learning techniques such as logistic regression, support vector machine (SVM), or

random forest to learn a model Zhang et al. (2019). The model can be then applied to make prediction for new data and be evaluated.

8.2 Datasets and experimental settings

We collect Nikkei 225 index[10] for a period of four months between June and September 2017. We use the closing price as the price of the day. We then make a time series of price movement where $r_i = p_i - p_{i-1}$. We create the training instances as described previously using r_i. We use the first three months of data as the training data, and data in the last month as the testing data.

We collect social media data using the method described in the previous section and align them with the stock market index. Then we generate context data from the social media data. We consider three ways of adding social media data as the contextual data:

- Bag-of-Words (BOW). The vector is simply a frequency count of each word. We consider all words that appear at least 5,000 times in the dataset. There are 760 such words. Each element of the vector for time t correspond to the frequency of the word within time t.
- Mean Word Vector (MWV). This method considers the semantics of words as represented in pre-trained word2vec vectors. As each word is associated with a word2vec vector, the vector for time t is the mean vector of vectors of all words appear in time t
- Event Representation (Event). We convert the events detected using our proposed method to temporal vectors. Each element in the vector is correspond to a dimension in the event data. Initially all elements in all vectors are assigned 0. Suppose dimension d is affected in an event happened between t_a and t_b. Then dimension d in all vectors for times between t_a and t_b is assigned either -1 or 1. It is assigned 1 if the average value of dimension d between t_a and t_b is higher than its average value in the entire time series. Otherwise it is assigned -1. So we consider not only the components and the duration, but also properties of the event. We will only test the retrospective event detection method, but based on earlier findings, we have reasons to expect the incremental method would perform similarly.

We measure the accuracy using *precision*, *recall* and *F1*, which are most used evaluation metrics in information retrieval Zhang et al. (2016). Precision measures how many correct positive predictions in all positive predictions, while recall measures the portion of actual positives being predicted. F1 is a balanced measurement based on precision and recall. We use SVM implemented in the R package e1071[11] as our supervised machine learning technique, although using other techniques shows similar tendencies.

8.3 Evaluation results

The prediction accuracy results of using only the historical price and three ways of adding social media context are shown in Table 5. We can see from the table that neither

[10] https://indexes.nikkei.co.jp/en/nkave/
[11] https://cran.r-project.org/web/packages/e1071/index.htm

Table 5 Stock market index prediction accuracy considering social media background

	Precision	Recall	F1
Historical Price	0.58	0.70	0.64
+ BOW	0.57	0.40	0.47
+ MWV	0.56	0.90	0.69
+ Event	0.69	0.90	0.78

BOW-based context nor the MWV-based context improves the prediction accuracy by much, compared to using just historical prices. However, using event-based context improves the accuracy significantly, with precision increased by 11% and recall increased by 20%. Thus it is evident that representing the social media temporal background using events is better than shallow representation such as BOW and word embeddings. The advantage comes from that the event method captures most unusual behavior in social media, and ignores the noises that may affect other presentations.

9 Conclusion

In this paper we propose a general method for event detection on social media. Two main steps of our method are generalizing social media text into word embeddings, and detecting multi-dimension events from time series. The detected events represent something unusual and affect semantic aspects of social media discussions, over a finite period. Compared to previous works on social media event detection, our method makes very few assumptions. We only assume that the event will be affecting a finite number of dimensions and, when affected, these dimensions behave differently from their usual, normal behavior. We propose two versions of the algorithm, a retrospective version and an incremental version. We evaluate proposed methods first on a synthetic dataset, then in two real-world computational tasks. With the synthetic dataset, we show that the incremental version can achieve almost the same performance as the retrospective version. In the first real-world task, we test detected events from social media discussions against three news categories, exhaustively collected over a testing period, and find that when the news is quite different from the base social media discussion, it can be better captured based on the detected events. We also found that the incremental version of the algorithm performs comparatively with the retrospective version. In the second real-world computational task of stock market movement detection, we show that representing social media as detected events has a clear advantage over other kinds of representation, in terms of improvement in prediction accuracy.

Despite some positive results from the indirect evaluation, we consider that our method has some drawbacks. For example, our method demands a test of normality, and requires a large portion of base data, which may not be always available. Furthermore, if it is a long period event, event-related semantics would become the norm and thus there would be a problem detecting the event with our method. Nevertheless, our method has its merits. It can be easily implemented and applied to more specific datasets. One can, for example, pre-select a discussion dataset about finance or entertainment, and apply our method to detect events of certain types. We can also make an event-based representation of social media that is much more efficient than heavy representations such as bag-of-words and word embedding, as we demonstrated in the second application

scenario. In the future, we plan to further develop the method to consider multimodal data, such as images and audio.

Acknowledgements This research is partially supported by JST CREST Grant Number JPMJCR21F2.

Availability of data and material The data used in this paper are available from the corresponding author upon request.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection in twitter. *Computational Intelligence, 31*(1), 132–164. https://doi.org/10.1111/coin.12017

Bartels, R. (1982). The rank version of von neumann's ratio test for randomness. *Journal of the American Statistical Association, 77*(377), 40–46. https://doi.org/10.1080/01621459.1982.10477764

Batal, I., Fradkin, D., Harrison, J., Moerchen, F., Hauskrecht, M. (2012). Mining recent temporal patterns for event detection in multivariate time series data. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 280–288. https://doi.org/10.1145/2339530.2339578

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science, 2*(1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

Cataldi, M., Di Caro, L., Schifanella, C. (2010). Emerging topic detection on twitter based on temporal and social terms evaluation. In: Proceedings of the Tenth International Workshop on Multimedia Data Mining, pp. 4:1–4:10. https://doi.org/10.1145/1814245.1814249

Chen, Y., Amiri, H., Li, Z., Chua, T. S. (2013). Emerging topic detection for organizations from microblogs. In: Proceedings of the 36th international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43–52. ACM. https://doi.org/10.1145/2484028.2484057

Cheng, H., Tan, P. N., Potter, C., Klooster, S. (2009). Detection and characterization of anomalies in multivariate time series. In: Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 413–424. SIAM. https://doi.org/10.1137/1.9781611972795.36

Dong, X., Mavroeidis, D., Calabrese, F., & Frossard, P. (2015). Multiscale event detection in social media. *Data Mining and Knowledge Discovery, 29*(5), 1374–1405. https://doi.org/10.1007/s10618-015-0421-2

Gao, Y., Wang, S., Padmanabhan, A., Yin, J., & Cao, G. (2018). Mapping spatiotemporal patterns of events using social media: a case study of influenza trends. *International Journal of Geographical Information Science, 32*(3), 425–449. https://doi.org/10.1080/13658816.2017.1406943

Guralnik, V., Srivastava, J. (1999) Event detection from time series data. In: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 33–42. https://doi.org/10.1145/312129.312190

Hua, T., Chen, F., Zhao, L., Lu, C. T., Ramakrishnan, N. (2013). Sted: semi-supervised targeted-interest event detectionin in twitter. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1466–1469. https://doi.org/10.1145/2487575.2487712

Khalifa, M. B., Diaz Redondo, R. P., Vilas, A. F., & Rodríguez, S. S. (2017). Identifying urban crowds using geo-located social media data: a Twitter experiment in New York City. *Journal of Intelligent Information Systems, 48*(2), 287–308. https://doi.org/10.1007/s10844-016-0411-x

Khodabakhsh, M., Kahani, M., & Bagheri, E. (2020). Predicting future personal life events on twitter via recurrent neural networks. *Journal of Intelligent Information Systems, 54*(1), 101–127. https://doi.org/10.1007/s10844-018-0519-2

Kim, J. (1976). Events as property exemplifications. In: Action Theory, pp. 159–177. Springer. https://doi.org/10.1007/978-94-010-9074-2_9

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery, 7*(4), 373–397. https://doi.org/10.1023/A:1024940629314

Li, R., Lei, K. H., Khadiwala, R., Chang, K. C. (2012). TEDAS: A Twitter-based event detection and analysis system. In: Proceedings of 28th International Conference on Data Engineering, pp. 1273–1276. https://doi.org/10.1109/ICDE.2012.125

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013) Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119

Minnen, D., Isbell, C., Essa, I., Starner, T. (2007). Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery. In: Proceedings of the Seventh IEEE International Conference on Data Mining, pp. 601–606. IEEE. https://doi.org/10.1109/ICDM.2007.52

Olteanu, A., Castillo, C., Diaz, F., Vieweg, S. (2014). CrisisLex: A lexicon for collecting and filtering microblogged communications in crises. In: In Proceedings of the 8th International AAAI Conference on Weblogs and Social Media, pp. 376–385

Parikh, R., Karlapalem, K. (2013). ET: events from tweets. In: Proceedings of the 22nd International Conference on World Wide Web, Companion Volume, pp. 613–620. ACM. https://doi.org/10.1145/2487788.2488006

Pennington, J., Socher, R., Manning, C. (2014). Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543

Popescu, A.M., Pennacchiotti, M. (2010). Detecting controversial events from Twitter. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1873–1876. https://doi.org/10.1145/1871437.1871751

Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. *Applied Intelligence, 26*(1), 25–33. https://doi.org/10.1007/s10489-006-0001-7

Ritter, A., Etzioni, O., Clark, S., et al. (2012). Open domain event extraction from twitter. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1104–1112. ACM. https://doi.org/10.1145/2339530.2339704

Rossi, C., Acerbo, F., Ylinen, K., Juga, I., Nurmi, P., Bosca, A., Tarasconi, F., Cristoforetti, M., & Alikadic, A. (2018). Early detection and information extraction for weather-induced floods using social media streams. *International Journal of Disaster Risk Reduction, 30*, 145–157. https://doi.org/10.1016/j.ijdrr.2018.03.002

Saeed, Z., Abbasi, R. A., Maqbool, O., Sadaf, A., Razzak, I., Daud, A., Aljohani, N. R., & Xu, G. (2019). What' s happening around the world? a survey and framework on event detection techniques on twitter. *Journal of Grid Computing, 17*(2), 279–312. https://doi.org/10.1007/s10723-019-09482-2

Sakaki, T., Okazaki, M., Matsuo, Y. (2010). Earthquake shakes Twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th International World Wide Web Conference, pp. 851–860. https://doi.org/10.1145/1772690.1772777

Sakaki, T., Okazaki, M., & Matsuo, Y. (2013). Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering, 25*(4), 919–931. https://doi.org/10.1109/TKDE.2012.29

Shoji, Y., Takahashi, K., Dürst, M.J., Yamamoto, Y., Ohshima, H. (2018). Location2vec: Generating distributed representation of location by using geo-tagged microblog posts. In: International Conference on Social Informatics, pp. 261–270. Springer. https://doi.org/10.1007/978-3-030-01159-8_25

Sul, H. K., Dennis, A. R., & Yuan, L. (2017). Trading on twitter: Using social media sentiment to predict stock returns. *Decision Sciences, 48*(3), 454–488. https://doi.org/10.1111/deci.12229

Suliman, A. T., Al Kaabi, K., Wang, D., Al-Rubaie, A., Al Dhanhani, A., Ruta, D., Davies, J., Clarke, S. S. (2016). Event identification and assertion from social media using auto-extendable knowledge base. In: Proceedings of 2016 International Joint Conference on Neural Networks, pp. 4443–4450. IEEE. https://doi.org/10.1109/IJCNN.2016.7727781

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826. https://doi.org/10.1257/mac.1.1.58

Taylor, J. B., & Williams, J. C. (2009). A black swan in the money market. *American Economic Journal: Macroeconomics, 1*(1), 58–83.

Unankard, S., Li, X., & Sharaf, M. A. (2015). Emerging event detection in social networks with location sensitivity. *World Wide Web, 18*(5), 1393–1417. https://doi.org/10.1007/s11280-014-0291-3

Vahdatpour, A., Amini, N., Sarrafzadeh, M. (2009). Toward unsupervised activity discovery using multi dimensional motif detection in time series. In: Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence

Walther, M., Kaisser, M. (2013). Geo-spatial event detection in the twitter stream. In: Proceedings of the 2013 European Conference on Information Retrieval, pp. 356–367. Springer. https://doi.org/10.1007/978-3-642-36973-5_30

Wang, Y., Jin, F., Su, H., Wang, J., Zhang, G. (2018). Reasearch on user profile based on user2vec. In: Proceedings of the 2018 International Conference on Web Information Systems and Applications, pp. 479–487. Springer. https://doi.org/10.1007/978-3-030-02934-0_44

Weng, J., Lee, B. S. (2011). Event detection in twitter. In: Proceedings of the Fifth International Conference on Weblogs and Social Media, pp. 401–408

Xie, W., Zhu, F., Jiang, J., Lim, E. P., & Wang, K. (2016). TopicSketch: Real-time bursty topic detection from twitter. *IEEE Transactions on Knowledge and Data Engineering, 28*(8), 2216–2229. https://doi.org/10.1109/TKDE.2016.2556661

Xu, Y., Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1970–1979. https://doi.org/10.18653/v1/P18-1183

Zhang, T., Zhou, B., Huang, J., Jia, Y., Zhang, B., Li, Z. (2017). A refined method for detecting interpretable and real-time bursty topic in microblog stream. In: Proceedings of the 2017 International Conference on Web Information Systems Engineering, pp. 3–17. Springer. https://doi.org/10.1007/978-3-319-68783-4_1

Zhang, Y., Maekawa, T., Hara, T. (2021). Using social media background to improve cold-start recommendation deep models. In: Proceedings of 2021 IEEE International Joint Conference on Neural Networks IJCNN, pp. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534327

Zhang, Y., Shirakawa, M., Hara, T. (2021). A general method for event detection on social media. In: Proceedings of the 25th European Conference on Advances in Databases and Information Systems ADBIS 2021. https://doi.org/10.1007/978-3-030-82472-3_5

Zhang, Y., Siriaraya, P., Kawai, Y., Jatowt, A. (2019). Analysis of street crime predictors in web open data. Journal of Intelligent Information Systems pp. 1–25. https://doi.org/10.1007/s10844-019-00587-4

Zhang, Y., Szabo, C., Sheng, Q. Z. (2016). Improved object and event monitoring on twitter through lexical analysis and user profiling. In: Proceedings of the 17th International Conference on Web Information System Engineering, pp. 19–34. https://doi.org/10.1007/978-3-319-48743-4_2

Zhang, Y., Szabo, C., Sheng, Q. Z., & Fang, X. S. (2018). SNAF: Observation filtering and location inference for event monitoring on twitter. *World Wide Web, 21*(2), 311–343. https://doi.org/10.1007/s11280-017-0453-1

Zhao, L., Chen, F., Lu, C. T., & Ramakrishnan, N. (2016). Online spatial event forecasting in microblogs. *ACM Transactions on Spatial Algorithms and Systems (TSAS), 2*(4), 1–39. https://doi.org/10.1145/2997642

Zhou, X., & Chen, L. (2014). Event detection over twitter social media streams. *The VLDB Journal, 23*(3), 381–400. https://doi.org/10.1007/s00778-013-0320-3