PAPER

Frame-Based Representation for Event Detection on Twitter

Yanxia QIN[†a)], **Yue ZHANG**[††], ***Nonmembers***, **Min ZHANG**[†††], ***Member***, ***and*** **Dequan ZHENG**[†], ***Nonmember***

SUMMARY Large scale first-hand tweets motivate automatic event detection on Twitter. Previous approaches model events by clustering tweets, words or segments. On the other hand, event clusters represented by tweets are easier to understand than those represented by words/segments. However, compared to words/segments, tweets are sparser and therefore makes clustering less effective. This article proposes to represent events with triple structures called frames, which are as efficient as, yet can be easier to understand than words/segments. Frames are extracted based on shallow syntactic information of tweets with an unsupervised open information extraction method, which is introduced for domain-independent relation extraction in a single pass over web scale data. This is then followed by bursty frame element extraction functions as feature selection by filtering frame elements with bursty frequency pattern via a probabilistic model. After being clustered and ranked, high-quality events are yielded and then reported by linking frame elements back to frames. Experimental results show that frame-based event detection leads to improved precision over a state-of-the-art baseline segment-based event detection method. Superior readability of frame-based events as compared with segment-based events is demonstrated in some example outputs.

key words: *frame, event representation, tweet, event detection, bursty, z-score*

1. Introduction

Social media provides an useful way for information dissemination. Different from traditional news media, social media enables the public to participate in information generation and transmission, even expressing opinions. Analyzing large scale real-time tweets for event detection assists public opinion monitoring, advertising and brand image maintaining etc. There is a rich body of work focusing on Twitter event detection, both supervised methods [1] and unsupervised methods [2]. We investigate an unsupervised framework for event detection on Twitter.

In this article, an **event** is defined as a collection of representation units, showing "what happened", and event detection in Twitter aims to find events from the stream of raw tweets. In previous research, events can be detected with different levels of granularity. Table 1 shows an example with different event representations. The most fine-grained level is *word level* [2]–[5], which represents events with highly event informative words selected from tweets. In Table 1, an event is represented by a set of anomalous high frequenty words. However, such independent word-based representation is always difficult to understand, and thus [6] propose a *segment level* event detection method, by extracting frequently used phrases and named entities as segments in tweet segmentation. However, the disadvantage of segment-based methods [6], [7] is still low readability without structured information on "who did what to whom" (i.e. "who win the game"). To address the low readability challenge, [8]–[10] propose the *tweet level* event detection methods by regarding each short tweet as one document, and fit it in traditional document-based clustering methods. However, they suffer from severe data sparseness and high time- and memory-cost issues given large tweet stream.

Table 1 A sugar bowl football match event "Louisville Cardinals 33-23 Florida Gators" in Jan. 2nd, 2013.

Unit	**Output event**
tweet	gameday! come on in later and watch the gators take down louisville! #sugarbowl
	rt @eancaafootball: retweet if your were impressed by louisville's huge upset over florida in the sugar bowl! http://t.co/vjqrccvu
word	*florida; bowl; sugar; louisville*
segment	*florida; sugar bowl; sec; bowl; louisville*
frame	**(louisville, gets biggest win in, program history)**; **(goes, florida)**; **(uf, was favored by, 14)** ; **(go, gators)**; **(go, cards)**

In this article, we propose *frame-based* event detection by considering the disadvantages of both word (segment) and tweet level methods. A **frame** is defined as a triple, denoted as $(arg_s, verb, arg_o)$, containing a verb phrase $verb$ representing an action and two noun phrases (i.e. arg_s and arg_o) representing the subject and object with respect to the action, respectively. By preserving subject and object information of tweets, frames are natural representation units encoded with structured information. Frames are extracted by considering the syntactic information of tweets, and hence are more semantically meaningful than words and segments. In addition, frame-based representation requires a basic degree of grammaticality of tweets, and thus can filter noisy tweets that are syntactically meaningless. The redundancy of tweets makes it possible by enabling each event has at least one related frame.

Compared to tweet-based methods, the proposed

Manuscript received September 26, 2017.
Manuscript revised December 5, 2017.
Manuscript publicized January 18, 2018.

[†] The authors are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China.

[††] The author is with the Information Systems Technology and Design, Singapore University of Technology and Design, 487372, Singapore.

[†††] The author is with the School of Computer Science and Technology, Soochow University, Suzhou, 215006, China.

a) E-mail: qolina@163.com

DOI: 10.1587/transinf.2017EDP7311

frame-based event detection method is more efficient because it applies burstiness-based filtering before clustering. Tweets are overly sparse for tweet-level bursty filtering. Readability of frame represented events remains comparable to events represented by tweets because important event information has been encoded in frames. As shown in Table 1, we can easily find that Twitter users are cheering for "florida", "gators" and "cards" and "louisville gets biggest win in the game" from frame-based event representation. Frame-based detection can capture more information compared with segment-based event representation, without losses any readability compared with tweet-based methods. An interesting observation is that people are shocked that louisville win the game as most of them think "gators take down louisville" before the football game, as stated in the first tweet in Table 1.

After obtaining frame-based representation of each tweet, we employ burstiness-based filtering to select more informative frames. Similar frames are clustered into groups, serving as events. In particular, *frame elements* (i.e. arg_s, *verb* and arg_o) in a frame are treated as intermediate processing units and being fitted in bursty element filtering and element clustering. For outputs, events represented by frame elements are further reported by frames through a linking procedure from frame elements to original frames. While words/segments are difficult for a linking process because they are more likely to appear in multiple events. The proposed **Fr**ame based representation for **E**vent **D**etection on Twitter, **FrED**, outperforms the segment-based method of Li et al. [6] (**Twevent**) on a benchmark of 31 million tweets.

The rest of the article is organized as follows. Section 2 introduces related work on event detection on Twitter. Section 3 gives an overview of FrED. Section 4 presents the frame-based representation method. Section 5 describes the proposed frame-based event detection method, including bursty frame element detection, element clustering and event filtering. Section 6 shows the event reporting method. Section 7 reports the experimental setting and result analysis. The article is concluded in Sect. 8.

2. Related Work

There are three typical stages for event detection in Twitter: 1) event representation; 2) event feature filtering; 3) event detection. In this article, we focus on event representation. Based on the level of granularity, existing work on event representation can be categorized into three categories namely feature-based models, tweet-based models, and structure based models.

Feature-based event representation includes word-based [3], [4], [11] and segment based [6], [7] models. Word-based methods use a cluster of similar words to represent events. Tweets are tokenized into words directly in [3]. Words co-occurred with named entities are selected as more representative words and taken as representation units [4]. Cui et al. [11] use popular hashtags to represent bursting events. Segments, proposed by [6], are supposed to be more meaningful than words, as they contain n-gram information. Tweets are separated into non-overlapping n-grams through an optimization process. Feature-based event representations can only present unstructured event information, which cannot directly show structured information "who did what to whom".

Tweet level event detection methods [8]–[10] use tweets for event representation. In [8], tweets are represented as vectors through bag-of-words models with TF-IDF weighting schema. Similarly, tweets are represented as a tweet-term matrix by a weighting method [9]. The proposed frame-based event representation provides comparably equal readability of events to tweet-based representation.

Besides representing events using flat documents, words or segments, other event detection methods [12]–[16] extract predefined event properties and organize them into structured events. Given a set of seed events, Benson et al. [12] extract artist and venue information of concerts. Popescu et al. [13] extract main entities, actions and audience opinions. These structured representations either focus on specific type of events for well-predefined event properties, or build structured events by linking separate event information through co-occurrence. The proposed structured frames are extracted directly from tweets by considering syntactic information of tweets, leading to higher accuracy.

There are also research focus on event feature filtering and clustering. For filtering, burstiness is a effective measurement to determine if a event is noise or not. Keyword-based filtering [17]–[19] is applied to find specific types of events. Besides keywords-based filtering, [9] also applied a structured tweet filtering considering the length of mentions and hashtags. Classification-based methods [10] are also used for tweet filtering. In this article, we use burstiness-based approach to filtering event features. In addition, various efficient clustering methods are explored. Petrovic et al. [20] propose to use Locality-Sensitive Hashing algorithm for first story detection from large scale tweets. Becker et al. [8] use a simple threshold based online clustering method. Ifrim et al. [9] utilize hierarchical clustering method for event detection. In this article, we use a simple but efficient k-Nearest Neighbor based graph partitioning method for clustering.

The effectiveness of using frames on event detection is also verified in our preliminary experiments in [21]. After replacing segments in Twevent [6] with extracted frame triples, we observed higher readability and precision. This work is a significant extension of [21]. This article develops a more general and effective bursty frame element detection method rather than the method in Twevent. This article also makes extensive comparisons with different baselines such as a supervised event representation method [14] and investigates the effect of a language model based preprocessing in helping frame-based event representation. Experimental settings in this work are improved than [21].

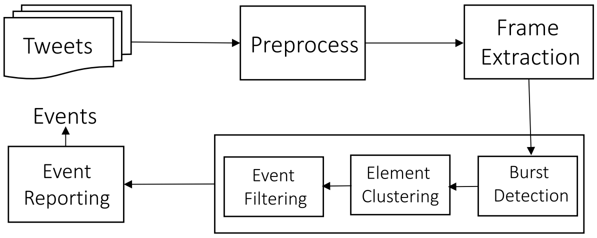

Fig. 1 Framework of frame-based event detection on Twitter.

3. Framework Overview

FrED consists of the following components: preprocessing, frame-based representation method, frame-based event detection (including bursty frame element detection, element clustering and event filtering) and event reporting. First, a fast preprocessing procedure is conducted over raw Twitter data. Then, frames are extracted based on syntactic information of tweets. Third, in event detection, bursty frame elements are grouped into clusters and trustworthy clusters that are able to represent events are reserved after an event filtering step. The main reason for not clustering frames directly is sparsity. The semantic structure nature of frames makes them overly sparse to be clustered, while frame elements serve as n-grams. Finally, for event reporting, frame elements from the resulting event clusters are linked back to their corresponding frames who are utilized to describe the events. The proposed framework is illustrated in Fig. 1.

4. Frame-Based Event Representation

4.1 Preprocessing

Raw Twitter data are very noisy and written in multiple languages, which motivates a preprocessing step. First, non-English tweets are removed by language detection [22]. Second, words in tweets are further normalized with a dictionary-based lexical normalization method [23], which replaces a word's lexical variants to its standard form. Third, part-of-speech analysis is conducted with a Twitter specific model [24] for syntactic analysis, and used for frame extraction. Finally, noun phrase identification [25] is applied for frame extraction.

4.2 Frame Extraction

As stated earlier, this article defines a frame as a triple (arg_s, *verb*, arg_o), where *verb* represents an action and arg_s and arg_o represent the verb's subject and object, respectively.

ReVerb [26], an open information extraction architecture, is taken for frame extraction. Rather than other tools [27], [28], ReVerb is adopted on Twitter data for the following reasons: first, it uses an unsupervised method, which is suitable for Twitter data due to fast-changing Twitter topics and extremely large scale of Twitter data. Second, ReVerb does not require syntactic parsing, which is still a very difficult task for Twitter data. While Twitter-specific shallow syntactic analysis like POS and chunking are mature enough to be utilized to extract frames through syntactic constraints in ReVerb.

As for *verb*, we focus on three main types of verb phrases, stated as follows, which includes a single verb (e.g. *smokes*), a simple verb phrase including a verb followed by a preposition (e.g. *come on*) and a complex verb phrase containing optional words (e.g. *gets biggest win in*).

$$\begin{aligned} & V/VP/VW^*P \\ & V = verb\ particle?adv? \\ & W = (noun/adj/adv/pron/det) \\ & P = (prep/particle/inf.\ marker) \end{aligned} \tag{1}$$

In practical, we obtain all possible matches for a verb phrase by regular expressions. The longest match is kept, and multiple matches are merged to one if they intersect with each other. Noted that, the lexical constraints, which is proposed in ReVerb to deal with over-specific verbs, are ignored in FrED because the following bursty frame element selection (Sect. 5.1) conducts even a more rigorous filtering. According to the fact that subject-verb-object (SVO)[†] is the main sentence structure in English, the nearest noun phrase to the left of each verb phrase is regarded as arg_s and the nearest one to the right as arg_o.

As a main contribution of our method, frame extraction filters out a large quantity of noisy words, which fail to satisfy the above syntactic constraints. Such words include misspelled words, user-defined words, abbreviations, emoticons and so on. They do not carry essential information about events, and are not included in the extracted frames. In Twevent [6], these words are filtered out by means of matching segments against Microsoft Web N-gram online service. Frame-based representation is independent of this resource, which makes the resulting frame elements highly effective for event detection.

4.2.1 Frame Examples

Table 2 presents some examples of frames/segments extracted from tweets. As shown in the Table, both the segment- and frame-based methods can extract meaningful segments or frames from tweets, respectively. In most cases, frames are more representative than segments by connecting verbs and their subjects/objects together (e.g. the frame in first tweet (phil taylor, announce, his retirement tonight)). In addition, NP-chunking results are more reliable than tweet segmentation. Taking the name in the third tweet "Demba Ba" as example, segmentation yields a "ba", while "demba ba" is identified as a noun phrase serving as an element in frame extraction. However, many long verb phrases in frames are also yielded, which causes a sparseness problem. Extracting more concise representation of frames will be one of our future directions.

[†] http://en.wikipedia.org/wiki/Subject-verb-object

Table 2 Example of different representations of tweet content (tweet/*segments*/**frames**).

Representation of Tweet Content
Justin bieber smokes weed?! Omg shocking *justin bieber; smokes; weed; shocking* **(justin bieber, smokes, weed)**
what price for phil taylor to announce his retirement tonight *price; phil taylor; announce; retirement; tonight* **(phil taylor, announce, his retirement tonight)**
demba ba look's class for chelsea exactly the kind of player we could do with at newcastle *ba; class; chelsea; exactly; player; newcastle* **(demba ba, look's class for, chelsea); (we, could do with at, newcastle)**
wake up arsenal and take a risk i believe that anyone at the club really believes that individual games i *wake; arsenal; take; risk; believe; club; believes; individual; games* **(-, wake up, arsenal); (arsenal, take, a risk); (i, believe that, anyone); (the club, really believes, that individual games)**

5. Frame-Based Event Detection

In a certain day d, given a set of tweets $T = \{t_1, t_2, \dots, t_n\}$, each tweet is a set of frames $t_i = \{f_1, f_2, \dots, f_m\}$, and each frame is structured as a triple with three elements $(args_s, verb, arg_o)$. Since the distributions of events and frames are sparse, we use the elements of each frames as the basic unit to detect if the corresponding tweet mentions a event. In particular, we first calculate the burstiness of each element to find the bursty elements, and then cluster these bursty elements through a k-Nearest Neighbor graph. Finally, event clusters are ranked and filtered by a heuristic-based method.

5.1 Element Filtering with Burstiness

The term **bursty** is used to refer to a feature's anomalous high frequent appearance in a time window over a time period. It is assumed that bursty features indicate appearance of important events. Bursty property has been widely adopted for identifying important event in a time series [4], [6], [29], [30], and we use burstiness to find event informative elements.

Given an element e (e could be $args_s$, arg_o or $verb$) and time window d, the probability of frequency of e in d is modeled by the following Gaussian distribution:

$$P(f_{e,d}) \sim \mathcal{N}(N_d p_e, N_d p_e(1 - p_e)), \tag{2}$$

where N_d is the number of tweets within time window d, and p_e is the expected probability of tweets containing e in a random time window:

$$p(e) = \frac{\sum_{d \in D} f_{e,d}}{\sum_{d \in D} N_d}, \tag{3}$$

where D is a long time period consisting several time windows. We then use z-score [31] to measure the burstiness of element e, which is defined in Eq. (4).

$$z(e, d) = \frac{f_{e,d} - E[e|d]}{\sigma[e|d]} \tag{4}$$

$z(e, d)$ can be used to measure the difference between the frequency $f_{e,d}$ and the expected value $E[e|d]$ (calculated as $N_d p_e$) in units of standard deviation $\sigma[e|d]$ (calculated as $\sqrt{N_d p_e(1 - p_e)}$). Top ranked elements by z-score are selected as bursty elements for further clustering.

5.2 Element Clustering

After obtaining the bursty elements, we use k-Nearest Neighbor graph (kNNgraph) to cluster them, and considering each cluster represent an event. kNNgraph, a variant of Jarvis-Patrick clustering algorithm [32] groups two elements into the same cluster only when they are each other's k-nearest neighbors. The value k determines both the number of clusters and the size of clusters, being set to 5 empirically.

For finding an element's k nearest neighbors, we need to calculate the similarity between two elements. A stream based model is used to calculate the similarity between two elements with temporal order. In particular, we firstly split a day d into m time windows as $< d_1 \dots d_M >$, and then calculate the similarity in each time window. Finally, we sum all the time window based similarity as final similarity between two elements. The similarity is calculated as:

$$sim(e_1, e_2) = \sum_{d_m} sim_{d_m}(e_1, e_2), \tag{5}$$

where sim_{d_m} is the similarity on time window d_m

$$sim_{d_m}(e_1, e_2) = w_{d_m}(e_1) w_{d_m}(e_2) cosine_{d_m}(T_1, T_2), \tag{6}$$

where $w_{d_m}(e) = f_{e,d_m}/f_{e,d}$ is the frequency weight of e in d_m, since we consider if two elements are similar, they should co-occur in the same time window. T_i is a set of tweets containing e_i within d_m, and T_i is represented with bag-of-words model and weighted by TF-IDF.

5.3 Event Filtering

After grouping all the elements into clusters, we need to use a heuristic *newsworthiness* score to filter them, since some clusters are personal updates or constant topics rather than news events [6]. We use two measurements to filter mundane clusters: 1) The probability of being a news for a cluster (P_{news}); 2) The cohesion score of a cluster (S_{coh}).

The probability of a cluster being a news P_{news} is calculated as:

$$P_{news} = \frac{\sum_{e \in S_c} \mu(e)}{|S_c|}, \tag{7}$$

where $S_c = \{e_1, e_2, \dots, e_n\}$ is a set of elements in cluster c, and $\mu(e)$ is the probability of element e being recognized as anchor texts in Wikipedia, defined as:

$$\mu(e) = \max_{l \in e} exp(Q(l)) - 1, \tag{8}$$

where l is sub-phrase of e and $Q(l)$ is the probability that l

appears as anchor text in Wikipedia. Feature with larger $Q(\cdot)$ can gain relatively higher $\mu(\cdot)$ by inducing the exponential function to boost the influence of $Q(\cdot)$.

The cohesion score S_{coh} is defined as:

$$S_{coh} = \frac{\sum_{e_a \in S_c} \sum_{e_b \in S_c - \{a\}} sim(e_a, e_b)}{|S_c|} \tag{9}$$

where the similarity between two elements $sim(e_a, e_b)$ is calculated using Eq. (5).

Finally, we combine P_{news} and S_{coh} as the *newsworthiness* of a cluster c:

$$\mu(c) = P_{news} \cdot S_{coh} \tag{10}$$

Hence, a cluster c is taken as a news event only if it satisfies the condition that $\mu_{max}/\mu(c) < \tau$, where τ is a threshold for *newsworthiness*, μ_{max} is the maximum value of $\mu(c)$ for all event clusters in time window d.

6. Event Reporting

After event filtering, we obtain a set of clusters $C = \{c_1, c_2, \ldots, c_n\}$, each cluster contains a sets of elements $c_e = \{e_1, e_2, \ldots, e_m\}$. Since there are many elements in a cluster, we only choose top k' elements with $\mu(e)$ score (Eq. (8)) to represent an event. In addition, since the readability of elements is lower than frames, we need to map the elements into frame. However, a element would map to many frames, we thus map e to the frame $F^*_{e,d}$, which contains e and has highest frequency in d. Hence a event cluster c can be represent as a set of frames $\{F^*_{e_1,d}, F^*_{e_2,d}, \ldots, F^*_{e'_k,d}\}$.

The advantage of frame-based event representation is that each frame is a meaningful semantic triple, while segment-based representation [6] is some independent n-gram phrases, which have no feasible linking-back schema to the original tweets.

7. Experiments

7.1 Data

Our Twitter data are crawled using Twitter public streaming API and consist of tweets published from Jan. 1st to Jan. 15th, 2013. The data set contains 31 million tweets published by 16 million users with 382 thousand words. A summary is shown in Table 3. Comparison of the average number of word, segment and frame per day gives a hint on sparseness of frame. In addition, Twitter data on Jan. 1st and Jan. 5th is regarded as a development set, and the rest 10 days' data is taken as the test set.

Wikipedia dump of Feb. 4th, 2013[†] is used as an extra resource for event filtering. These entities' anchor probabilities (i.e. the number of pages on which entity e appears as anchor text divided by the number of pages containing the entity e) are calculated in event filtering. It includes 13 million pages and 10 million anchor entities that have the 5 word length limit.

Table 3 Data statistics.

Unit	Average(/day)	Total
tweet	2, 073 K	31, 097 K
word	79 K	382 K
segment	288 K	1, 604 K
frame	1, 797 K	14, 948 K
frame element	1, 439 K	14, 957 K

7.2 Experimental Settings

7.2.1 Baseline Bursty Feature Detection Methods

To evaluate the effectiveness of z-score based bursty detection method, we compare with bursty detection method used in Twevent [6]. For fairness, different bursty detection methods are fit into Twevent system for comparison.

Twevent: bursty feature selection method in [6], denoted as **B**ursty probability and **U**ser frequency, BU-based feature selection. All the symbols are the same as Sect. 5.1. Heuristic BU-based feature selection considered two factors: *bursty probability* ($P_b(e, d)$), which shows how bursty the feature e is in day d, and *user frequency* ($u_{e,d}$), the number of users that tweeted about the feature, indicating how popular the feature is within Twitter users.

$$w_b(e, d) = P_b(e, d) log(u_{e,d}) \tag{11}$$

$u_{e,d}$ is used to filter out noisy features, as the more users talk about e, the more popular and meaningful it is. $u_{e,d}$ is calculated as the number of users who post tweets containing e within time window d.

$P_b(e, d)$ is the bursty probability calculated in Eq. (12). Specially, $P_b(e, d)$ is defined as 1 when $f_{e,d} >= E[e|d] + 2\sigma[e|d]$. $S(\cdot)$ is the sigmod function.

$$P_b(e, d) = S(10 \times \frac{f_{e,d} - (E[e|d] + \sigma[e|d])}{\sigma[e|d]}) \tag{12}$$

Twevent$_z$: the proposed z-score based feature selection method in Sect. 5.1,

Twevent$_{zu}$: another system by combining z-score and the user frequency part of Eq. (11). Features are ranked by z-score and $u_{e,d}$ in Eq. (11), respectively, and those that ranked highly in both list are taken as bursty features.

In all related methods, the time window d is set to one day and each time window is divided into $M = 12$ sub time windows in clustering. k in kNNgraph clustering method is set to 5 emprically.

7.2.2 Baseline Event Detection Methods

To compare event detection methods with different representation units, we compare our methods with different event detection methods.

Twevent: Twevent [6] is taken as one of the baselines. We do not compare with word-based methods, given that

[†] http://burnbit.com/download/235406/enwiki_20130204_pages_articles_xml_bz2

segment-based methods outperform word-based event detection methods.

FrED$_{evt}$: Similar to TwiCal [14], which extracts event phrases and named entities for event representation, we build another baseline FrED$_{evt}$. In FrED$_{evt}$, frames are constructed by taking each event phrase as a verb phrase *verb*, the nearest named entity to the left as arg_s and the nearest named entity to the right as arg_o. (arg_s, *verb*, arg_o) is taken as one frame if arg_s or arg_o is not empty. Named entities and event phrases are extracted from tweets using the tool published by [14]. Different from FrED, in which event phrases are verb phrases, event phrases in FrED$_{evt}$ can be event-related verbs, nouns and adjectives, which are recognized through a linear chain Conditional Random Fields (CRF) model. Comparison of FrED and FrED$_{evt}$ gives us a hint on how different frame extraction methods influence event detection performance.

FrED$_{filt}$: To investigate whether the quality of tweets influences the performance, another baseline, FrED$_{filt}$ is designed, which uses an open source US English Language Model† to filter out low-score tweets as a preprocessing step.

7.3 Evaluation

Precision and events number are used as evaluation metrics. In addition, two annotators are asked to evaluate the experimental results manually. Output events of the group of feature selection experiments are evaluated by two annotators, and Cohen's Kappa is applied to calculate agreement of the two annotators. Results of the second group of experiments are randomly assigned to one of two annotators. An event is represented by a given date and a group of features (e.g. segments for Twevent, frames for FrED). Annotators are asked to judge whether the event is a news event which happened on the given date. News that happened before the given date can also be annotated as true news event, as some events can stay hot in tweets for several days. This work regards what really happened as news events, including sports news, entertainment news, technical news etc. Search engines are allowed to assist annotating, with selected features and the given date as queries.

7.4 Results and Analysis

This section shows the experimental results of all the experiments. It also presents some example events of Twevent and FrED showing their event readability.

7.4.1 Evaluation on Bursty Feature Selection Methods

The results of different bursty feature selection methods are shown in Table 4. #Event is the number of events detected by systems, replacing incomputable recall indicator. #AgrEvt is the number of events whose labels are agreed by two annotators. Precision is the number of true events out of all agreed events, computed as #trueEvent/#AgrEvent. Kappa is Cohen's kappa value of two annotators' labeling results.

Table 4 Experimental results of bursty feature selection methods on develop set.

System	#Event	#AgrEvt	Precision	Kappa
Twevent	51	43	67.44%	0.65
Twevent$_z$	42	37	**72.97%**	0.71
Twevent$_{zu}$	24	22	68.18%	0.81

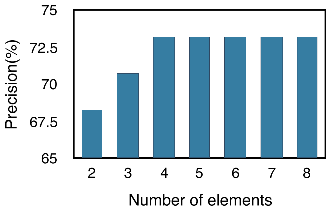

Fig. 2 Effect of different number of frame elements of FrED on dev data.

A Cohen's kappa value, over 0.6, by the three methods indicates that human agreement on event detection is high, since it has been received that 0.6 is a threshold of Cohen's kappa denoting acceptable agreement. Higher value means higher agreement. Twevent$_z$ performs the best in precision among all three methods, which verifies that the z-score based method is superior than the baseline method in identifying bursty features. The reason that Twevent$_z$ outperforms Twevent$_{zu}$ is that the function of z-score and user frequency overlaps. Features with higher z-scores also indicate popularity between Twitter users. A decreased number of events (#Event) in Twevent$_z$ and Twevent$_{zu}$ can be an indicator for stronger constraint of these feature selection methods. Accordingly, the z-score based method is used for bursty feature selection in following experiments.

An examination shows that most segments in Twevent differs very little in bursty probability $P_b(e,d)$. It means bursty segments yield from Twevent are ranked only by user frequency, which leads to worse performance than Twevent$_z$.

7.4.2 Influence of Different Number of Frame Elements for Event Representation

In event reporting (Sect. 6), we select top k' meaningful frame elements to report events. While too less frame elements may leads to incomplete information, and too much elements may add noisy information. We conduct an experiment on FrED system on development data to investigate the effect of different number of elements on event detection. We change k' from 2 to 8 based on a statistical analysis, in which we found the number of frame elements in events ranges from 2 to 8 with the averaged number be 4. The results are shown in Fig. 2.

Shown in Fig. 2, the precision of FrED with selected top 2 or 3 elements is lower than others. After investiga-

†http://cmusphinx.sourceforge.net/2013/01/a-new-english-language-model-release/

Table 5 Experimental results of different event detection methods on test data.

System	#Event	Precision
Twevent$_z$	107	66.36%
FrED	62	**70.97**%
FrED$_{evt}$	70	65.31%
FrED$_{filt}$	49	70%

Table 6 Example events with different location of informative frame. FrED_ele is FrED without event reporting, whose output event is represented by frame elements. Frame elements in FrED_ele are highlighted in frames in FrED. Event 1: A Football game, with the forth frame to be most informative[†]. Event 2: A false multi-event.

ID	System	Event
1	FrED_ele	averaging; 56; 4 tds; has played in
	FrED	(4 tds, **averaging**, 56); (4 tds, averaging, **56**); (**4 tds**, averaging, 56); (thomas, **has played in**, 5 quarters)
2	FrED_ele	corinthians; boateng; pato; ac milan
	FrED	(**corinthians**, have confirmed the signing of, alexandre pato); (**boateng**, pick, the ball); (transfers; confirmed; **pato**); (**ac milan**; have done more to tackle racism in; one day)

Table 7 Example output events of Twevent, FrED_ele and FrED. Event 1: football game (Florida vs Louisville). Event 2: football game (Arsenal vs Southampton).

ID	System	Event
1	Twevent	florida; sugar bowl; sec; bowl; louisville
	FrED_ele	florida; louisville; uf; gators; cards
	FrED	(-, goes, **florida**); (**louisville**, gets biggest win in, program history); (**uf**, was favored by, 14); (-, go, **gators**); (-, go, **cards**)
2	Twevent	arsenal; southampton; 1-1; walcott; ramsey
	FrED_ele	gervinho; southampton; arsenal; rvp
	FrED	(-, you're a better footballer than, **gervinho**); (-, done, **southampton**); (-, come on, **arsenal**); (**rvp**, don't think, he)

tion, we found this is caused by some events with 2 or 3 frames could not provide enough information to recognize the events. For example, first event in Table 6, the first three frame elements link to one same meaningless frame. The most event informative frame is the forth one.

In addition, the precision of FrED stabilize after we select 4 elements to report events, which means at least 4 elements could generally represent full information. We select k' to be 5 in following experiments. To be noted that we label those events including multiple event information to be false events. Intuitively, those multi-events may be labeled as true when use less elements to report events, and labeled as false when use more elements. Interestingly, we found those multi-events are recognized as false even use 2 elements. We assume this can be caused by high efficient element ranking algorithm in Sect. 5.3. A multi-event is shown in second event in Table 6, where the first and third frame indicate one football player Alexandre Pato's transferring event, while the second and forth one show event "AC Millan's Boateng stand against racism after walking off in protest at abuse".

7.4.3 Final Performances of FrED

The experimental results of FrED and baseline systems are presented in Table 5. Here precision and number of events (#Event) are used to evaluate the systems, with recall being replaced by the total number of detected events. This is because it is difficult to identify all events that happen over a period.

By using Twevent$_z$ and FrED, a contrast can be made between segment- and frame-based news event detection with same bursty feature selection method. Improvement on precision (66.36→70.97) verifies the effectiveness of frame-based event detection method compared to segment-based method. One of the main reasons for the improvement is that frame detection conducts feature selection by filtering out irrelevant non-frame words. In contrast, segment-based method relies on bursty feature detection to filter out infrequent phrases, without a refined feature selection step. For example, frequent words like 1) emoticons such as '<33333333' (love) and '555555' (crying); 2) onomatopoeia words such as 'hahahaha' (laughing) and 'hmmmm' (doubting or hesitating); 3) misspelled words such as 'restrictiv' (restrictive) and 4) meaningless words like 'xxxxxx' are considered by Twevent.

Compared to FrED, FrED$_{evt}$ performs worse in precision and better in event number. This is likely because error propagation from named entity extraction and event phase extraction on tweets should count for the precision loss. FrED$_{filt}$ yields comparably good precision as FrED, with less events being detected. It is believed that language model filtering is not helpful for improving the precision because of effectiveness of the bursty feature extraction.

#Event is influenced by the threshold τ, which evaluates difference between a cluster's *newsworthiness* and the largest *newsworthiness* of all candidate clusters. Since τ is fixed to 2 following Twevent, #Event can show distribution of *newsworthiness* between all clusters. A large #Event by Twevent shows clusters' quality differs from each others. Otherwise for FrED. Though, this article focus on the improvement of precision and readability of FrED.

Comparison of #Event between methods FrED$_{evt}$, FrED$_{filt}$ and FrED gives a hint on the similarity of *newsworthiness* of event clusters. More detected events in FrED$_{evt}$ show that event clusters detected have similar *newsworthiness* value than those in FrED. While event clusters in FrED$_{filt}$ have more decentralized *newsworthiness* value.

7.5 Example Output

Table 7 shows some event outputs, in which FrED gives more readable output summaries than Twevent. Note that there may not be corresponding arg_s or arg_o for a *verb* in one frame. Events detected by Twevent and the frame el-

[†] Here, 'tds' means touchdowns, a football terminology.

ement clusters of FrED (before event reporting) are mostly described by noun phrases without verbs, which can show important action information. In contrast, FrED could describe events with the frames, which contain verb phrases.

For the first event in Table 7, the resulting segments of Twevent are mostly participants of the event (i.e. *florida, louisville*). Different from Twevent, frames by FrED contain not only participants but also verb phrases showing the cheering action (i.e. *go/goes*). Frames in FrED suggest that users prefer to cheer for their favorite team rather than only stating the fact that which team wins. This observation serves as an evidence of the informal writing in social media messages. Frame *"(-, you're a better footballer than, gervinho)"* in second event gives a hint of sports news.

Another interesting observation is that sports and entertainment news take a large fraction of all resulting events. We assume that general Twitter users prefer to talk about sports and entertainment news. Further investigation may include detecting interests of Twitter users general topics.

8. Conclusion

In the proposed framework, a frame based representation method and a general bursty detection method for the event detection task on Twitter are developed. Different from words/segments, frames are structured information units and hence convey more event information. Compared with word and segment based methods, frame based methods has two advantages. First, frames naturally give readable summaries of events. Second, frame extraction requires relatively grammatical tweets, and therefore serves to filter noise and mundane tweets. Redundancy of tweets makes this feasible. Experiments show the effectiveness of the frame-based method through an improved precision over baseline systems.

Acknowledgments

We would like to thank anonymous reviewers' suggestions. We would like to thank Jiayuan Chao and Junjie Yu for helping annotate the experimental results. The work is supported by the National Natural Science Foundation of China (No. 61525205).

References

[1] X. Zhou and L. Chen, "Event detection over twitter social media streams," The VLDB Journal, vol.23, no.3, pp.381–400, 2014.
[2] L.M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Göker, I. Kompatsiaris, and A. Jaimes, "Sensing trending topics in twitter," IEEE Transactions on Multimedia, vol.15, no.6, pp.1268–1282, Oct. 2013.
[3] S. Lee, S. Lee, K. Kim, and J. Park, "Bursty event detection from text streams for disaster management," Proceedings of WWW Companion, Lyon, France, pp.679–682, 2012.
[4] A.J. McMinn and J.M. Jose, "Real-time entity-based event detection for twitter," Proceedings of CLEF, Toulouse, France, vol.9283, pp.65–77, Sept. 2015.
[5] M. Platakis, D. Kotsakos, and D. Gunopulos, "Searching for events in the blogosphere," Proceedings of WWW, Madrid, Spain, pp.1225–1226, 2009.
[6] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," Proceedings of CIKM, Maui, Hawaii, USA, pp.155–164, 2012.
[7] Y. Qin, Y. Zhang, M. Zhang, and D. Zheng, "Feature-rich segment-based news event detection on twitter," Proceedings of the Sixth IJCNLP, Nagoya, Japan, pp.302–310, Oct. 2013.
[8] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," Proceedings of ICWSM, 2011.
[9] G. Ifrim, B. Shi, and I. Brigadir, "Event detection in twitter using aggressive filtering and hierarchical tweet clustering," Proceedings of he SNOW 2014 Data Challenge, Seoul, Korea, pp.33–40, April 2014.
[10] R. Li, K.H. Lei, R. Khadiwala, and K.C.-C. Chang, "Tedas: A twitter-based event detection and analysis system," Proceedings of ICDE, Arlington, VA, USA, pp.1273–1276, 2012.
[11] A. Cui, M. Zhang, Y. Liu, S. Ma, and K. Zhang, "Discover breaking events with popular hashtags in twitter," Proceedings of the 21st CIKM, Maui, Hawaii, USA, pp.1794–1798, 2012.
[12] E. Benson, A. Haghighi, and R. Barzilay, "Event discovery in social media feeds," Proceedings of ACL-HLT, Portland, Oregon, pp.389–398, 2011.
[13] A.-M. Popescu, M. Pennacchiotti, and D. Paranjpe, "Extracting events and event descriptions from twitter," Proceedings of WWW, Hyderabad, India, pp.105–106, 2011.
[14] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from twitter," Proceedings of the 18th ACM SIGKDD, pp.1104–1112, 2012.
[15] D. Zhou, L. Chen, and Y. He, "A simple bayesian modelling approach to event extraction from twitter," Proceedings of ACL, Baltimore, Maryland, pp.700–705, Association for Computational Linguistics, June 2014.
[16] Y. Choi, P.M. Ryu, H. Kim, and C. Lee, "Extracting events from web documents for social media monitoring using structured svm," IEICE Trans. Inf & Syst, vol.E96-D, no.6, pp.1410–1414, 2013.
[17] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," Proceedings of WWW, Raleigh, North Carolina, USA, pp.851–860, 2010.
[18] P. Agarwal, R. Vaithiyanathan, S. Sharma, and G. Shroff, "Catching the long-tail: Extracting local news events from twitter," ICWSM, pp.379–382, 2012.
[19] Y. Kitagawa, M. Komachi, E. Aramaki, N. Okazaki, and H. Ishikawa, "Disease event detection based on deep modality analysis," Proceedings of the ACL-IJCNLP 2015 Student Research Workshop, Beijing, China, pp.28–34, Association for Computational Linguistics, July 2015.
[20] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," Proceedings of HLT, Los Angeles, California, pp.181–189, 2010.
[21] Y. Qin, Y. Zhang, M. Zhang, and D. Zheng, "Semantic-frame representation for event detection on twitter," Proceedings of IALP, 2017.
[22] M. Lui and T. Baldwin, "langid.py: An off-the-shelf language identification tool," Proceedings of the ACL 2012 System Demonstrations, pp.25–30, 2012.
[23] B. Han, P. Cook, and T. Baldwin, "Automatically constructing a normalisation dictionary for microblogs," Proceedings of the EMNLP-CoNLL, pp.421–432, 2012.
[24] L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva, "Twitter part-of-speech tagging for all: Overcoming sparse and noisy data," Proceedings of RANLP, 2013.
[25] L. Ramshaw and M. Marcus, "Text Chunking Using Transformation-Based Learning," Proceedings of the Third Workshop on Very Large Corpora, Somerset, New Jersey, pp.82–94, 1995.
[26] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," Proceedings of EMNLP, pp.1535–

1545, 2011.

[27] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni, "Open language learning for information extraction," Proceedings of the EMNLP-CoNLL '12, pp.523–534, 2012.

[28] G. Angeli, M.J. Premkumar, and C.D. Manning, "Leveraging linguistic structure for open domain information extraction," Proceedings of the ACL-IJCNLP, Beijing, China, pp.344–354, Association for Computational Linguistics, July 2015.

[29] A. Guille and C. Favre, "Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach," Social Network Analysis and Mining, vol.5, no.1, 2015.

[30] G.P.C. Fung, J.X. Yu, P.S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," Proceedings of VLDB, Trondheim, Norway, pp.181–192, 2005.

[31] E. Kreyszig, H. Kreyszig, and E.J. Norminton, Advanced Engineering Mathematics, 10th ed., Wiley, Hoboken, NJ, 2011.

[32] R.A. Jarvis and E.A. Patrick, "Clustering using a similarity measure based on shared near neighbors," IEEE Trans. Comput., vol.C-22, no.11, pp.1025–1034, Nov. 1973.

Yanxia Qin is currently a Ph.D. student of Harbin Institute of Technology. Her advisor is Prof. Min Zhang. She received her B.S. from Northeast Normal University and M.E. from Harbin Institute of Technology in 2010 and 2012, respectively. She interned at Institute for Infocomm Research in 2012, under supervision of Prof. Min Zhang. She interned at Singapore University of Technology in 2013 and 2017 twice, under supervision of Prof. Yue Zhang. Her research interests include natural language processing, deep learning, information extraction and social network.

Yue Zhang is currently an assistant professor at Singapore University of Technology and Design. Before joining SUTD in July 2012, he worked as a postdoctoral research associate in University of Cambridge, UK. Yue Zhang received his DPhil and MSc degrees from University of Oxford, UK, and his BEng degree from Tsinghua University, China. His research interests include natural language processing, machine learning and artificial Intelligence. He has been working on statistical parsing, parsing, text synthesis, machine translation, sentiment analysis and stock market analysis intensively. Yue Zhang serves as the reviewer for top journals such as Computational Linguistics, Transaction of Association of Computational Linguistics (standing review committee) and Journal of Artificial Intelligence Research. He is the associate editor for ACM Transactions on Asian and Low Resource Language Information Processing. He is also PC member for conferences such as ACL, COLING, EMNLP, NAACL, EACL, AAAI and IJCAI. He was the area chairs of COLING 2014, NAACL 2015, EMNLP 2015, ACL 2017 and EMNLP 2017. He is the TPC chair of IALP 2017.

Min Zhang is a distinguished professor in the School of Computer Science and Technology, Soochow University, Suzhou. He received his Bachelor degree and Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, in 1991 and 1997, respectively. His current research interests include machine translation, natural language processing, and artificial intelligence.

Dequan Zheng is currently an associate professor at Harbin Institute of Technology. He received his Bachelor degree from Heilongjiang University in 1991. He received his Master and Ph.D. degrees from Harbin Institute of Technology in 1998 and 2006, respectively. His research interests include data mining, information extraction and applied artificial intelligence. He is a member of Chinese Information Processing Society (CIPS) and China Computer Federation. He has won 1 Ministry Science and Technology Award and owned 1 national patent. He has hosted more than 10 projects like NSFC normal and the project of National High Technology Research and Development Program of China. He has published more than 80 papers on international journals and conferences.