BEHIND: a 4W-oriented Method for Event Detection from Twitter

Kang Zeng, Yujia Liu, Xin Song, and Bin Zhou*
College of Computer, National University of Defense Technology
Changsha, China
Email: {zengkang, liuyujia, songxin, binzhou}@nudt.edu.cn

***Abstract*—Event detection from Twitter has attracted attention from researchers in the past decade due to the widespread use of social media. By leveraging the knowledge derived from these events, it is possible to understand what consumers are interested in and give the opportunity for organizations to make better decisions. Numerous studies have proven the advantage of burst detection methods in detecting events in Twitter streams. However, some burst detection methods mainly focus on the bursty characteristics caused by events while the elements in events are not fully utilized. In this paper, we focus on the elements in *When, Where, Who*, and *What (4W)* dimensions of events and propose a *4W*-oriented event detection method called BEHIND. BEHIND jointly uses *Bursty Element*s and Heterogeneous Information Network(HIN) for event detection. *Bursty Element*s are calculated through probability distribution and they are used to select tweets with bursty elements. HIN is used to enhance relevance judgment in *4W* dimensions between tweets to help cluster tweets. The tweet clusters are corresponding to events we detected. We used a benchmark dataset to evaluate our method. Experimental results demonstrate that our method achieves higher precision and less duplication rate, and detects more events than the state-of-the-art methods.**

***Index Terms*—data mining, Twitter, event detection, event summarization, 4W**

I. INTRODUCTION

Recently, social media has overtaken print media as the main source of news gathering for consumers [1]. In the past decade, automatically detecting events from Twitter has attracted much attention from researchers.

Burst detection methods [2] [3] [4] [5] have been widely studied and applied for Twitter event detection. They focus on detecting events with bursty characteristics (i.e., breaking news). Those methods can help track topics of general interest and detect events in early stages. However, they still introduce some issues that can affect the results of event detection. We present two cases as examples of such issues as follows.

Case 1: *There were two events happened at the same time: "Chinese author Mo Yan won 2012 Nobel Prize in Literature" and "The European Union was awarded the 2012 Nobel Prize in Peace". Both were hotly discussed on Twitter. However, some methods reported an event about "Mo Yan" and "The European Union", while there was no event happening between them at that time.*

Case 2: *There were three presidential debates between Barack Obama and Mitt Romney in 2012. Some methods all reported them as "presidential debate". Those reports could not identify which presidential debate it is.*

The first case shows that some burst detection methods confuse two co-occurring and related events, thus reporting an event that is not actually happened at all. The second case shows that these methods cannot discriminate between multiple occurrences of the same type of event.

In order to get more interpretable event detection results, [6] and [7] borrowed the definition of events from journalism. They defined social media events as: a social media event can be represented by *When, Where, Who* and *What (4W)* dimensions. This definition can help identify an event by multiple different aspects of information, while it is generic enough to generalize most social media events. Nevertheless, it is still not fully utilized in Twitter event detection. Some methods [7] using this definition simply incorporate features in 4W dimensions into their process.

To better deal with the issues that may be caused by burst detection methods and better use of the 4W representation, we propose a *4W*-oriented (*When*, *Where*, *Who*, *What*) method called BEHIND for event detection. BEHIND jointly uses ***Bursty Element*s** and **Heterogeneous Information Network**(HIN) [8] to **Detect** events in Twitter stream. We firstly extract elements in *4W* dimensions from tweets. Then we select *bursty element*s of each dimension and use them to filter tweets. This can filter out noisy data in early stages and improve the precision of event detection. *Bursty Element*s will be discussed in more detail in Section III-A2.

In addition, we build a HIN on the filtered tweets. The nodes in HIN are tweets, and the types of edges in HIN include *When, Where, Who* and *What (4W)*. We discuss this in Section III-B for more details. We use HIN to help reconstruct the feature representation of tweets to strengthen the connection between tweets based on elements in *4W* dimensions. We use *Tweet Clustering based on HIN* to make it easier to cluster tweets discussing the same event into the same cluster. *Tweet Clustering based on HIN* will be discussed in Section III-C. This not only reduces repeated reports of the same event, but also increases the possibility of detecting more events. Then we cluster tweets based on the new feature representation. The resulting clusters are the events we detected.

Finally, we concatenate the top elements in *4W* dimensions

*Corresponding author
DOI reference number:10.18293 / SEKE2021-092

of those events to generate event summaries. The summaries can make the detected events more interpretable and discriminative. Because elements in *4W* dimensions can jointly verify an event and present more comprehensive information about the event.

To sum up, the contributions of our work are:

- We propose a *4W*-oriented (*When*, *Where*, *Who*, *What*) method called BEHIND to detect events in the irregular text stream of tweets.
- BEHIND jointly uses *bursty element*s and HIN to detect events. The former is used to effectively select event tweets, while the latter is used to enhance relevance judgment in *4W* dimensions between tweets for clustering.
- Comparative experiments show that BEHIND can achieve higher precision and less duplication rate, and can detect more events.
- The events detected by BEHIND are summarized by the top elements in *4W* dimensions, which make the detected events more interpretable and discriminative.

II. Related Work

Twitter event detection has been extensively studied over the past decade. For the purpose of better understanding the existing related work, we classify the existing event detection methods based on the common traits they share.

A. Bursty-term-based Methods

Generally speaking, the occurrence of an event always triggers people to discuss it in Twitter. A relevant number of methods detected events by extracting bursty terms from tweets and clustering such terms to get events. Twevent [9] and SEDTWik [3] used tweet segment, which is defined as one or more consecutive words appearing in a tweet to help detecting events. They extracted bursty tweet segments and clustered them to get events. TopicSketch [2] relied on the concept of word acceleration to detect trending topics on Twitter. It calculated the occurrence rate of pairs or triples of words as the word velocity. The change in velocity within the two time windows is calculated as the acceleration.

However, the textual contents of tweets are sparse and informal, detecting events by bursty terms may detect clusters of terms that are weakly correlated with realistic events.

B. Social-aspect-based Methods

The way people discuss interesting events on Twitter is much more different from the way people share things in their daily lives. Social aspect information can be utilized for event detection. [10] built the relationship between tags to get a graph of related tags and detected bursty tagging events by extracting subgraphs. Generative Latent Dirichlet Allocation Model (MGe-LDA) [11] is a hashtag-based Mutual for detecting events in Twitter. MGe-LDA emphasized the role of hashtags in the semantic representation of the corresponding tweets. MABED [12] is a statistical method that relied solely on the creation frequency of user mentions that users insert into the tweets to detect important events.

Event detection methods focus on the social aspects of Twitter may only detect the most influential events and ignore the small-scale events. Meanwhile, they may require more hyper-parameter, such as the number of top events to detect.

C. Entity-based Methods

Entities are always considered to contain great event information and can help detecting events more efficiently. [13] examined the roles of entities on event detection. They partitioned and clustered documents based on the entities which contained to represent an event. [7] defined semantic categories based on *4W* dimensions, which included named entity, mention, location, hashtag, verb, noun and embedded link. They aggregated tweets discussing the same event into one cluster by the similarity measure between those semantic categories. [14] used entities on Twitter Trends to help clustering and used entity clusters to represent events. It addressed scaling issues with new design choices that link event clusters and enable real-time event detection through evolutionary tracing.

Those entity-based methods usually require lots of computational resources and labeled data. furthermore, most of them did not make full use of the textual semantic features of tweets.

III. Methodology

Fig. 1 shows the architecture of BEHIND. It consists of four components: data processing, tweet HIN building, tweet clustering and event summarization.

A. Data Processing

1) Elements Extraction: We filter tweets by *bursty element*s in *4W* dimensions. This can greatly reduce the computational cost and improve precision of event detection. We use a few advanced NLP tools[12] to extract relevant elements from tweets. We consider "time" extracted from tweets as elements in *When* dimension, "country" and "location" as elements in *Where* dimension, "person", "organization" and "@username" as elements in *Who* dimension. Generally speaking, elements in *What* dimension are very diversified. Inspired by the concept of *text segment* from [9] and [3] which refer to one or more consecutive words, we use text segments to represent the *What* dimensional elements of Twitter events.

2) Tweets Filtering by Bursty Elements: Thousands of tweets are generated every day, and most of them (i.e., spam, self-promotion, pointless babble) do not contain information to help event detection. Therefore, after extracting event elements, we calculate the *bursty element*s that may be related to events and discard the remaining ones.

Bursty Element

[9] introduced the concept of *bursty segment* to detect Twitter events. *Bursty segment* refers to one or more consecutive words that abnormally burst in tweets within a time window. We only consider the elements in *4W* dimensions of events and we only extract the burst elements in *4W* dimensions, which

[1] https://github.com/OpenSextant/Xponents
[2] https://github.com/FraBle/python-sutime

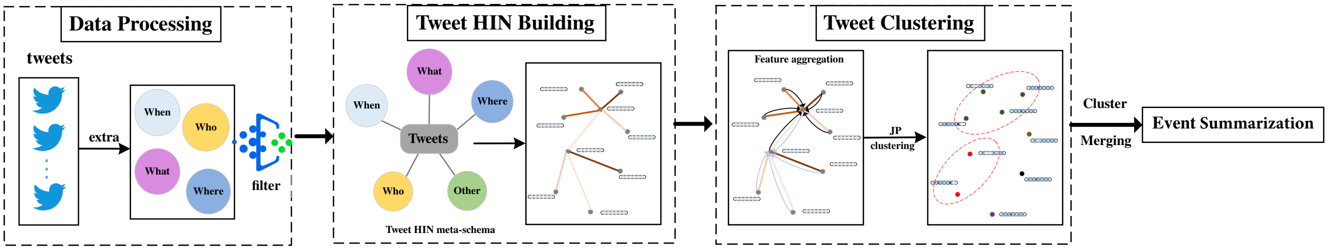

Fig. 1. An overview of BEHIND: Input data consists of Twitter data sorted by time. We extract elements in *4W* dimensions from input data and calculated the bursty elements to filter the event tweets. Then, we build a HIN with edges as elements in *4W* dimensions of filtered tweets. We use the initial embedding generated by BERT and feature aggregation based on HIN to generate new embedding of tweets. We cluster tweets through new embeddings to get events. Finally, we extract the top elements in *4W* dimensions in event clusters to get the event summaries.

alleviate the misleading effect of useless *bursty segment*s (e. g., thank god, every day).

Let N_T denotes the total number of tweets and $f_{ele,T}$ denotes the number of element ele in time window T. $f_{ele,T}$ can be considered as a Binomial distribution $B(N_T, p_{ele})$ and p_{ele} denotes as the expected probability of ele in any random time window. Since N_T is large enough, it can be considered that $E[ele|T] = N_T p_{ele}$ and $\sigma[ele|T] = \sqrt{N_T p_{ele}(1-p_{ele})}$. We use a formula for the bursty probability $P_b(ele,T)$ for ele in time window T defined by [9] as given in (1), where $S(\cdot)$ is the sigmoid function.

$$P_b(ele,T) = S(10\frac{f_{ele,T} - (E[ele|T] + \sigma[ele|T])}{\sigma[ele|T]}) \tag{1}$$

Taking into account the social aspect of Twitter, $src_{ele,T}$ denotes the sum of retweet count of all tweets containing ele in T and $u_{ele,T}$ denotes the number of users who use the ele in T. Both of them also affect the precision of event detection.

Finally, we define the bursty weight of the element ele as:

$$w_b(ele,T) = P_b(ele,T)log(u_{ele,T}) \times log(src_{ele,T}) \tag{2}$$

We sort the elements by their bursty weights. The top $\sqrt{N_T}$ elements in each dimension are called *bursty element*s.

After repeatedly comparative experiments, we only keep tweets containing at least two dimensional *bursty element*s, which can use reasonable computing resources to achieve great event detection results. Besides, N_f denotes as the number of remaining tweets.

B. Tweet HIN Building

We use Heterogeneous Information Network (HIN) in BEHIND to enhance relevance judgment between tweets for clustering. Here we introduce some basic definitions based on previous work [8].

Definition 3.1 Heterogeneous Information Network (HIN) *A Heterogeneous Information Network (HIN) is a graph $G = (V, E)$ with a object mapping function $\Phi : V \to A$ and a link mapping function $\varphi : E \to R$ while the type of objects $|A| > 1$ or the type of relations $|R| > 1$. V denotes the object set ,A denotes the object type set, E denotes the link set and R denotes the link type set.*

Definition 3.2 Meta-schema *Given a HIN G, the meta-schema $T_G = (A, R)$ for G is a graph with nodes as object types from A and edges as relations type from R.*

We show an example of the HIN meta-schema in Fig. 1. The object type in A is tweet and the relation types in R include *4W*(*When*,*Where*,*Who*,*What*) and Other(e. g., social aspect relations).

Definition 3.3 Meta-path *Meta-path $\mathcal{P}$ is defined on the network schema $T_G = (A, R)$, the specific form is:*$\mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2} \cdots \xrightarrow{\mathcal{R}_l} \mathcal{A}_{l+1}$.

The meta-path $\mathcal{P}$ defines a combination relationship $R = R_1 \cdot R_2 \cdot \cdot R_l$ between node types A_1 and A_{l+1}, while $\cdot$ denotes the combination operation between relations.

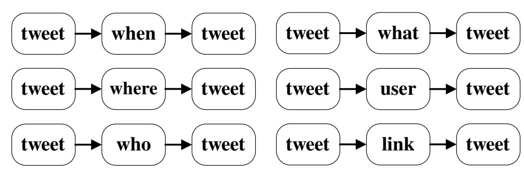

Fig. 2. Example of Meta-paths

According to the definitions given above, we use the filtered tweets to build a tweet HIN. We show a few meta-paths instances in Fig. 2. For example, tweet t_i and tweet t_j contain the same element in *4W* dimensions or contain the same link, these can be used to establish meta-paths between t_i and t_j.

C. Tweet Clustering based on HIN

After the HIN is built, we reconstruct the feature representation of tweets to better cluster tweets discussing the same event. We introduce pre-trained BERT [15] embeddings as the initial embeddings of tweets. We define the initial embedding of tweet t_i as h_i.

1) Feature Aggregation: We reconstruct the embeddings of tweets by feature aggregation in HIN. We use a path-count [16] strategy as the initial similarity measure of two tweets in HIN, which is the number of meta-paths between tweet i and tweet j: $e_{i,j} = |p : p \in \mathcal{P}|$.

Moreover, if there are meta-paths between two tweets, the two tweets are neighbors of each other. Note that tweet itself is also its own neighbor. The neighbors of t_i is defined as:

$$\mathcal{N}_i = \{t_j | e_{i,j} > 0\} \tag{3}$$

Finally, we aggregate the features from $\mathcal{N}_i$ to tweet t_i through the normalized similarity measure, so that we get the new embedding z_i of tweet t_i:

$$z_i = \sum_{j \in \mathcal{N}_i} \frac{e_{i,j}}{\sum_{k \in \mathcal{N}_i} e_{i,k}} \cdot h_j \tag{4}$$

2) Jarvis-Patrick (JP) Clustering: After getting the new embeddings of tweets, we can get the final similarities between all tweets and all their neighbors by cosine similarities. Then we sort them to get the k-nearest neighbors of each tweet.

Finally, all tweets can be clustered by JP algorithm [17]. In this, we treat all tweets as separate nodes initially, an edge is added between tweet t_i and tweet t_j if k-nearest neighbors of t_i contains t_j and vice versa. After traversing all nodes, all connected components can be considered as candidate event clusters in time window T, and the remaining nodes without any edges are discarded.

3) Cluster Merging: We extract *4W* dimensional elements from tweets in the candidate event clusters. Some candidate clusters without elements in *When*, *Where* and *Who* dimensions are discarded to get better results. To better manage these candidate event clusters, we not only query the event clusters of the current time window T, but also query the event clusters of the time window $T-1$. We compare the elements in *When*, *Where* and *Who* dimensions of each two clusters. The *What* dimension are not considered here because the number of elements in the *What* dimension is usually too large. If the elements coincidence rate is greater than 50%, we merge these two events. The remaining clusters are the events BEHIND finally detected.

TABLE I
RESULTS OF BEHIND AND BASELINES

	No. event	Precision	DERate
MABED	21	74.00	71.62
SEDTWik	28	70.59	22.22
BEHIND-noAgg	31	82.16	**18.42**
BEHIND-tfidf	36	72.58	20.00
BEHIND	**52**	**83.12**	18.75

D. Event Summarization

Reasonable event summaries can be used to query and manage events. Existing Twitter event detection methods mainly use a keyword set [9] [14] or a representative tweet [18] to describe an event. The former may generate some keyword sets that are not associated with one realistic event. The latter also struggle to find a representative tweet that sums up the whole event due to the brevity of tweets.

We support that a Twitter event may have corresponding *4W* dimensional elements. These elements in different dimensions can jointly identify an event, and make events more effectively queried and managed. Therefore, we sort the count of elements in each dimension of those events. Then, we select the top three elements of each dimension for an event and concatenates them to describe the event.

For example, we use *"2012-10-17 | us, america, new york | mitt romney, obama | debate, middle class, president obama"* to summarize the event of *"Second Presidential Debate between Obama and Romney in 2012"*. This allows us to observe that there is a debate between Barack Obama and Mitt Romney in New York on October 17, 2012.

IV. EVALUATION

A. Dataset and Setup

1) Dataset: To evaluate the performance of BEHIND, we use a huge Twitter dataset called Events2012 [19] to evaluate BEHIND. The entire dataset includes 120 million tweets. Since Events2012 only contains tweet IDs, we use a crawler to get this corpus. Meanwhile, the results of Twitter event detection need to be manually examined, we confirm that it would cost a lot of time to use the entire data set for experiments. For the time constraint and the volume limitation of Twitter, we use the corpus from October 11 to October 17 in Events2012 to evaluate BEHIND in our work.

2) Baselines:

- **SEDTWik**: SEDTWik [3] is an extension of Twevent [9]. SEDTWik identifies event based on bursty segments and clusters these segments to get the important events. Experiments in [3] have shown that SEDTWik achieves better results than Twevent.
- **MABED**: MABED [12] is a method of event detection using social aspect feature, which is based on mention anomaly to detect events.
- **BEHIND-noAgg**: It is a variant of BEHIND, which removes the feature aggregation module and uses the initial embedding generated by BERT for clustering.
- **BEHIND-tfidf**: It is a variant of BEHIND, which uses TF-IDF instead of BERT pre-training model to generate the initial embedding of tweets. Most of event detection methods use TF-IDF for their clustering module.

3) Experimental Setup: We use three metrics to evaluate results of event detection, which are Number of events (*No. events*), *Precision*, Duplicate Event Rate (*DERate*). all three metrics are referred from [3] and [9] .

- **No. events**: the number of detected events that can be correlated with realistic events.
- **Precision**: the percentage of detected events that can be correlated with realistic events.
- **DERate**: the percentage of repeated detected events among all realistic events detected.

For the proposed BEHIND, we remove all retweets from the Twitter stream. Meanwhile, we set a time window T to be 24 hours, which can be adjusted according to the number of tweets. For the initial embeddings of tweets, we use the BERT model trained by Sentence Transformers [20], which are tuned specifically meaningful sentence embeddings such that sentences with similar meanings are close in vector space. We set the k used in Jarvis-Patrick algorithm as $N_f/1000$ to get the best experimental results.

TABLE II
A SAMPLING OF EVENTS DETECTED AND SUMMARIED BY BEHIND AND BASELINES

Event	Event detected by BEHIND, MABED and SEDTWik
New music video by Justin Bieber and Nicki Minaj performing Beauty And A Beat	2012-10 | canada | justin bieber, nicki minaj | beauty beat, music video, youtube –**BEHIND** video, youtube, beat, justin (bieber, amp, liked, playlist, i'm, uploaded, favorited, added, ass, music) –**MABED** good morning, youtube, vp debate, justin bieber, beauty and a beat video, paul ryan,mitt romney, joe biden, thank god, tcot –**SEDTWik**
Red Bull Stratos	NULL | stratos | red bull, youtube, felix baumgartner | edge space, red bull stratos, liked video –**BEHIND** livejump (redbullstratos, jump, space, baumgartner) –**MABED** felix baumgartner, red bull stratos, livejump, stratos, edge space, record breaking, thank much, try best, felix, apple maps –**SEDTWik**
Second Presidential Debate between Barack Obama and Mitt Romney	2012-10-17 | us, america, new york | mitt romney, obama | debate, middle class, president obama –**BEHIND** question, debates, romney (president, ask, answering, don't, amp, answer, obama, mitt) –**MABED** (Not Detected) –**SEDTWik**
Cowboys vs Ravens on Oct 14, 2012	2012-10-14 | dallas, baltimore, detroit | dallas cowboys, baltimore ravens, fox | field goal, baltimore ravens, dallas cowboys –**BEHIND** cowboys (ravens, game, lose, win, fan, dallas, fuck) –**MABED** (Not Detected) –**SEDTWik**
Hilary Mantel's novel Bring Up the Bodies won the 2012 Booker Prize for the second time	2012 | NULL | hilary mantel, booker prize | man booker prize, bring bodies, second time –**BEHIND** (Not Detected) –**MABED** milk, news, breaking news, rt two, gary mckinnon, male thoughts, man booker prize, fox news, wise man, hilary mantel –**SEDTWik**
Space Shuttle Endeavour Embarks on L.A. Road Trip	2012-10-12 | los angeles, wells, fargo | angeles, wells fargo, branch manager | los angeles ca, space shuttle, greater los angeles –**BEHIND** (Not Detected) –**MABED** (Not Detected) –**SEDTWik**
Chinese author Mo Yan wins Nobel Prize in Literature	2012-10-11 | academy of sciences, china | mo yan, swedish academy, nobel prize | nobel prize literature, nobel literature prize, christian science –**BEHIND** (Not Detected) –**MABED** (Not Detected) –**SEDTWik**
Megan Fox Gives Birth to First Child With Brian Austin Green	2012-09 | NULL | megan fox, brian, megan | birth baby, first child, american actress –**BEHIND** (Not Detected) –**MABED** (Not Detected) –**SEDTWik**

For both SEDTWik and MABED, we use the implementation provided by the authors. The number of top events to be detected is the hyper-parameter of MABED. We set it to 100, which get the best experimental results.

B. Result

1) Event Detection Results: All methods follow experimental setting in IV-A3. Specifically, we used Google News and Wikipedia Page Titles datasets to identify an realistic event. The detailed comparison is shown in table I. From the comparison results, we have the following observations and analyses:

- BEHIND achieves the best performance in *No. event* and *Precision* metrics and second-best performance in *DERate* metric, which shows BEHIND can cluster tweets discussing the same event better.
- MABED has weak performance in the experiments. By analyzing the results of MABED, we find that most of the results are related the event of *"Second Presidential Debate between Barack Obama and Mitt Romney"* and it always has been one of the most discussed events. Once a hot event occurs, MABED may not be able to detect other smaller-scale events that occur at the same time.
- Despite using the same experimental setup, SEDTWik does not perform as good as in [3]. We suggest that because some tweets cannot be crawled anymore, which affects the results of event detection. [21] also agrees with it, they reported that about 50% of tweet relevance judgments were deleted in Events2012. This also demonstrates BEHIND's ability to capture relevant judgments.

2) Ablation Experiments: Through the results of variants of BEHIND, we can get the following observations and analyses:

- On the whole, Both BEHIND-noAgg and BEHIND-tfidf perform worse than BEHIND. Nevertheless, they perform overall better than SEDTWik and MABED.
- BEHIND-tfidf achieves the second-best performance in the *No. event* and *DERate* metrics, while the result is poor in the *Precision* metric. We observe some results of BEHIND-tdidf, which aggregates the elements of different events into a cluster. This shows that the initial embedding generated by BERT can be better used to capture semantic relations between tweets in large corpus.
- BEHIND-noAgg achieves poor performance in *No. Events* metric. This shows that HIN can help capture more relevance judgment between tweets. It brings closer the representation of two tweets that describe different aspects of the same event, though not nearly as similar semantically.

3) Event Summarization: We show a sampling of results detected in Table II. The first column is the manually labeled events and the second column is the automatically generated event summaries by BEHIND, MABED and SEDTWik.

Event summaries generated by MABED include main words and common words. MABED assigns weights to those common words. To make the summaries more concise, we remove the weights of them. For each event in SDETWik, we only take the top ten segments as the event summary. Note that "Not Detected" means that the method does not detect this event, "NULL" in the event summaries generated by BEHIND means that elements of this dimension is not detected.

Next, we discuss two examples in Table. II to better demonstrate that our method can get interpretable and discriminative event summaries. The event of "*Chinese author Mo Yan wins Nobel Prize in Literature*" detected by BEHIND is summarized by "*2012-10-11 | academy of sciences, china | mo yan, swedish academy, nobel prize | nobel prize literature, nobel literature prize, christian science*". The summary gives a more holistic picture about an event by elements in 4W dimensions. MABED did not report this event, while SEDTWik only reported a result that contained "*Mo Yan*", "*EU*" and a few irrelevant segments. This corresponds to Case 1 of Section I.

The event of "*Second presidential debate between Obama and Romney in 2012*" detected by BEHIND is summarized by "*2012-10-17 | us, america, new york | mitt romney, obama | debate, middle class, president obama*". SEDTWik does not report this event, while MABED reports it as: "*question, debates, romney (president, ask, answering, don't, amp, answer, obama, mitt)*". The summary in MABED cannot determine which presidential debate it is, but the summary in BEHIND can confirm that this is the second presidential debate held in New York through "*2012-10-17*" and "*new york*". This corresponds to Case 2 of Section I.

From results in Table I and Table II, we argue that using *4W* dimensional elements for event detection can both improve detection performance and make detected events more interpretable and distinguishable. While due to the sparsity and irregularity of tweets, we can observe that the elements of some dimensions are missing. But in most cases, the given elements about an event have been able to identify and understand the event.

V. Conclusion

Twitter event detection has attracted great interests from both academia and industry. In this paper, we proposed a method called BEHIND for detecting events from Twitter, which mainly included filtering tweets by *burst element*s and clustering tweets based on HIN to figure out most relevant events. The evaluation based on benchmark dataset shows that BEHIND achieved higher precision and less duplication rate, and detects more events than the state-of-the-art methods. Meanwhile, BEHIND can derive interpretable event summaries. For potential future work, we consider using natural language generation to improve the readability of event summaries.

Acknowledgment

This work is supported by the National Key R&D Program of China under Grants (No.2018YFC0831703).

References

[1] "Social networks finally bypassed print newspapers as a primary source of news," 2018. [Online]. Available: https://www.adweek.com/digital/social-networks-finally-bypassed-print-newspapers-as-a-primary-source-of-news

[2] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang, "Topicsketch: Real-time bursty topic detection from twitter," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2216–2229, 2016.

[3] K. Morabia, N. L. B. Murthy, A. Malapati, and S. Samant, "Sedtwik: segmentation-based event detection from tweets using wikipedia," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, 2019, pp. 77–85.

[4] D. Paul, Y. Peng, and F. Li, "Bursty event detection throughout histories," *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1370–1381, 2019.

[5] C. Comito, A. Forestiero, and C. Pizzuti, "Bursty event detection in twitter streams," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, pp. 1 – 28, 2019.

[6] X. Chen and Q. Li, "Event modeling and mining: a long journey toward explainable events," *The VLDB Journal*, vol. 29, no. 1, pp. 459–482, 2020.

[7] Q. Li, A. Nourbakhsh, S. Shah, and X. Liu, "Real-time novel event detection from social media," in *2017 IEEE 33Rd international conference on data engineering (ICDE)*. IEEE, 2017, pp. 1129–1139.

[8] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2016.

[9] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 155–164.

[10] J. Yao, B. Cui, Y. Huang, and Y. Zhou, "Bursty event detection from collaborative tags," *World Wide Web*, vol. 15, no. 2, pp. 171–195, 2012.

[11] C. Xing, Y. Wang, J. Liu, Y. Huang, and W.-Y. Ma, "Hashtag-based sub-event discovery using mutually generative lda in twitter," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[12] 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, p. 18, 2015.

[13] A. J. McMinn and J. M. Jose, "Real-time entity-based event detection for twitter," in *International conference of the cross-language evaluation forum for european languages*. Springer, 2015, pp. 65–77.

[14] M. Fedoryszak, B. Frederick, V. Rajaram, and C. Zhong, "Real-time event detection on social data streams," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2774–2782.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] Y. Sun and J. Han, "Mining heterogeneous information networks: Principles and methodologies," in *Mining Heterogeneous Information Networks: Principles and Methodologies*, 2012.

[17] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on computers*, vol. 100, no. 11, pp. 1025–1034, 1973.

[18] M. Hasan, M. Orgun, and R. Schwitter, "Real-time event detection from the twitter data stream using the twitternews+ framework," *Inf. Process. Manag.*, vol. 56, pp. 1146–1165, 2019.

[19] A. J. McMinn, Y. Moshfeghi, and J. M. Jose, "Building a large-scale corpus for evaluating event detection on twitter," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 409–418.

[20] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[21] Ø. K. Repp, "Event detection in social media-detecting news events from the twitter stream in real-time," Master's thesis, NTNU, 2016.