

KEvent – A Semantic-Enriched Graph-Based Approach Capitalizing Bursty Keyphrases for Event Detection in OSN

Sielvie Sharma Muhammad Abulaish, *SMIEEE* Tanvir Ahmad
Department of Computer Engineering *Department of Computer Science* *Department of Computer Engineering*
Jamia Millia Islamia *South Asian University* *Jamia Millia Islamia*
New Delhi, India New Delhi, India New Delhi, India
sielvie@outlook.com abulaish@ieee.org tahmad2@jmi.ac.in

Abstract—Social networks are growing quickly, and they have soon taken over as the main global source of breaking news. As a result, these platforms provide a plethora of user-generated content, which has inspired researchers to delve into and interpret data for a variety of objectives. Due to its effectiveness in locating news items hidden inside enormous amounts of voluminous data, *event detection* in online social network data has recently grown in prominence. In this research, we introduce **KEvent**, a novel graph-based technique for event detection from Twitter messages (*aka* tweets). The suggested method divides tweets into bins for extracting bursty keyphrases and then uses post-processing techniques to create a weighted keyphrase graph using the **Word2Vec** model. The keyphrase graph is then subjected to Markov clustering for the purpose of clustering and event detection. **KEvent** is evaluated over the **Events2012** benchmark dataset, and it performs noticeably better when compared to two state-of-the-art techniques, **Twevent** and **SEDTwik**. Additionally, **KEvent** has the ability to find events that the aforementioned state-of-the-art techniques were unable to find.

Index Terms—Social Network Analysis, Event Detection, Bursty Keyphrase, Markov Clustering.

I. INTRODUCTION

Due to exponentially increasing popularity, online social network (OSN) serves as a primary source for reporting a majority of global events. Its striking features like dissemination of information and liberty to read and write have changed the whole idea of news distribution. Various platforms on the Web, such as Twitter, Facebook, and Instagram offer these services. Out of these OSN platforms, Twitter is widely used and it primarily contains short textual data, called *tweets*, comprising of maximum 280 characters. It is expanding at a rate of 30 percent every year¹ and has grown drastically over a short period of time due to its user-friendly features, such as hashtag (#), mention (@) and retweet (RT). Another component contributing to its popularity is the follower-

followee network, where a user may follow any other user for updates.

Twitter, an interactive OSN platform, enables the propagation of a rich and continuous data flow. Obtaining such data has gotten simpler with the recent developments in Web technologies. Twitter offers APIs for retrieving public data and expanding its influence through research. The ubiquitous availability of such helpful information encourages research ideas; and event detection is one such expanding concept. A social media event can be defined as an unusual outpouring of tweets by diverse users about a single subject within a specific time. The concept of event detection in Twitter involves processing enormous tweets to gather knowledge about existing noteworthy incidents. Information acquired from such events makes determining users interests and actions easier, allowing us to make informed decisions.

Various event detection approaches based on classification and clustering techniques are available in the literature [1][2]. However, most previous methods concentrated solely on the arrangement of words, such as frequencies and co-occurrences. The semantic-enriched approach is required to explore the second aspect of linguistics, i.e., semantics, given the structural ambiguity in OSN data. Single event information is derived from multiple tweets sent by numerous users. With current approaches available, two tweets cannot be from the same event if the word set of the tweets is not similar. Integration of semantics in such a situation can be a decisive factor. In addition, many previous studies employed supervised and semi-supervised algorithms that rely on seeding keywords that are known prior to detection. However, they fail to detect novel events for which a set of keywords are not known a priori.

In order to address the above-mentioned limitations, in this paper we propose a novel approach, called **KEvent**, for event detection in Twitter. It uses temporal equi-width

¹<https://www.internetlivestats.com/twitter-statistics/>

binning and post-processing methods to split tweets into bins for extracting bursty keyphrases. Thereafter, it generates a weighted keyphrase graph using the `Word2Vec` model. Finally, Markov clustering is applied over the keyphrase graph for clustering and event detection. In summary, the key contributions of this paper can be summarized as follows:

- 1) *Coping with voluminous data*: A temporal binning method to initially distribute tweets into equi-width bins and then follow a linear chaining approach to identify consecutive bins that are contextually similar and merge them together.
- 2) *Feature extraction and burst detection*: A keyphrase extraction approach to identify bursty keyphrases from each bins locally to conceptualize the events described by the tweets contained therein.
- 3) *Bursty graph generation and event detection*: A graph-based data modeling approach to model bursty keyphrases into a weighted graph and applying Markov clustering over it for event detection.

II. RELATED WORK

With the boom of OSN, event detection has become a significant research problem due to the exponential growth of user-generated data. In [3] and [4], the authors introduced social media, Twitter and event detection as a research challenge. Diverse authors perspectives on event detection resulted in the demonstration of different methodologies. Following are the two broad categorizations we discovered in the literature: The first is unspecified event detection in which voluminous data is available and information about the data is not known prior to the detection. This entails discovering previously unknown unique events. Unsupervised approaches are commonly utilized in unspecified event detection. A brief review of the literature associated with event detection in Twitter is presented in the following paragraphs.

The authors of [1] and [2] discussed various approaches that have been used in recent past for specified and unspecified event detection. They also highlighted significant problems and research difficulties, such as no particular performance evaluation metrics due to various detection methods. In [5], the authors used a segment-based approach wherein segment extraction is dealt as an optimization problem with the help of external sources like Microsoft web N-gram. Thereafter bursty segments are obtained by utilizing Twitter features like tweet frequency and user diversity combined with a segment probability distribution. The authors of [6] extended the above work by exploring additional features such as retweet count and follower count of segment contained in a tweet along with mentioned bursty probability for burst extraction. McMinn et al. [7] created a massive corpus of 120 million tweets that enclosed data for over

a month. With the help of Wikipedia and other cluster summarization approaches, 500 events are extracted and classified into eight categories.

All of the above-mentioned methods are based on the idea of unspecified event detection and so involve unsupervised methods. Labeling data is a complex process with the dynamic and constantly emerging social media data. Due to this, incremental clustering is popular among authors as there are no prerequisites while dealing with unknown data. However, they all employed only syntactical and statistical features and lacked semantics. Semantics are incorporated in previous researches, although in different forms, such as context prediction and rule-based approaches. The authors in [8] described an event as a composition of who, where, what and when? Proper nouns, mentions, location and hashtags are included as semantic features. Further, rule-based approaches are used for extracting and categorizing terms.

In contrast to unspecified event detection, another is specified event detection, where event information is known in advance. It could be planned events such as political and entertainment events. Supervised techniques are used for such type of scenarios [8][9]. Primary attention is on features that are domain dependent, such as the extraction of tweets with the help of hashtags or keywords related to an event. Due to the use of known seed keywords, supervised techniques cannot identify unanticipated events. As a result, supervised approaches are not ideal for handling challenges like event detection in online social networks.

In summary, integration of semantics in an appropriate way is rarely observed in the above literature regarding event detection. The proposed model attempts to fill this gap for efficient detection of events in Twitter.

III. PROPOSED APPROACH

This section presents a detailed description of our proposed approach, `KEvent`, which consists of various steps such as *temporal binning*, *feature identification and bursty keyphrase extraction*, and *bursty keyphrase graph generation and event detection*. Figure 1 presents the architecture of `KEvent` and table I presents a list of symbols used in this paper along with their brief descriptions. Further details about the individual functioning modules are presented in the following sub-sections.

A. Temporal Binning

Existing approaches divide Twitter data into random and, most popularly, hour intervals to handle such a prodigious amount of data. Unlike conventional arrangements, we present a method to define the division by preserving both content and context of the tweets. This module is designed to make computations easier and to

TABLE I: Symbols and their descriptions

Symbol	Description
$C(\text{SuperCorpus})$	Corpus of all tweets of a day
$N_{\text{intervals}}$	Total number of time intervals
T_i	The i^{th} time interval
$TB(\text{Tweets}(T_i), \text{Tweets}(T_{i+1}))$	Temporal binning for two consecutive intervals T_i and T_{i+1}
$\text{Hash}(T_i)$	Distinct hashtags belong to time interval T_i
$HC(T_i, T_{i+1})$	Hashtag convergence for two consecutive intervals T_i and T_{i+1}
$\text{Cat}(T_i)$	Distinct categories belonging to time interval T_i
$OC(\text{Cat}(T_i, T_{i+1}))$	Overlap coefficient for categories between two consecutive intervals T_i and T_{i+1}
$CCS(\text{Cat}(T_i, T_{i+1}))$	Common category score for categories between two consecutive intervals T_i and T_{i+1}
L	Total number of overlapping categories between two consecutive intervals T_i and T_{i+1}
$\text{Score}(\text{Cat}_K(T_i))$	Score of category 'K' in time interval T_i
$CC(T_i, T_{i+1})$	Categorical convergence for two consecutive intervals T_i and T_{i+1}
$MC(T_i, T_{i+1})$	Mutual convergence for two consecutive intervals T_i and T_{i+1}
$\text{Bin}(T_{\text{new}})$	A bin with new time interval
$NV(KP_a)$	Nearest Vocab of the bursty keyphrase KP_a
$EV(KP_a)$	Embedded Vector of the bursty keyphrase KP_a
$OC(NV(KP_a), NV(KP_b))$	Overlap Coefficient of nearest vocab of two bursty keyphrases
$\text{CosineSim}(EV(KP_a), EV(KP_b))$	Cosine similarity of the <code>word2vec</code> embeddings of the bursty keyphrases KP_a and KP_b
$\text{CombinedSim}(KP_a, KP_b)$	Combined similarity between the keyphrases KP_a and KP_b

capture the events that produce bursts for a brief period. Such bursts tend to dissipate or spread across multiple time periods with random time interval arrangements, further preventing keyphrases from manifesting as bursty keyphrases. This module also aids in the discovery of sub-topics associated to significant events, such as a singer's performance in a match. In order to illustrate the binning process, we have considered one-day data disseminated in 24-hour intervals by default to find appropriate bins.

1) **Hashtag Convergence:** It aims to find the common hashtags between the consecutive time intervals. In previous researches, vocabulary change was proposed and the term vocabulary is defined as the set of all the words in an interval [10]. Incorporating all the words of a time interval is a complex task and does not add much to the functionality as different users pick different sets of words to write about the same event. It is also stated that practically Twitter content corresponding to the same event may be less similar in words [11]. However, there are very few hashtags associated with an event compared to the vocabulary of words and provide much information in a very brief form [6]. Therefore, considering hashtags instead of whole word vocabulary is a more rational decision. To this end, hashtag convergence is calculated as defined in equation 1.

$$HC(T_i, T_{i+1}) = \frac{|\text{Hash}(T_i) \cap \text{Hash}(T_{i+1})|}{|\min(|\text{Hash}(T_i)|, |\text{Hash}(T_{i+1})|)} \quad (1)$$

In this equation, the numerator shows the number of common hashtags in consecutive time intervals T_i , T_{i+1} , and denominator represents the minimum number of distinct hashtags in both the intervals.

2) **Categorical Convergence:** When it comes to merging similar social media content together, content similarity is not enough. Unlike traditional media, social networking users describe an event in a very unstructured way in the form of tweets. To this account, categorical convergence along with hashtag convergence is combined to capture all the aspects of the tweets. Natural Language Understanding (NLU) is used to extract categories from the data. NLU is a service provided by IBM which uses deep learning techniques to extract meaning and metadata from unstructured text data. Keywords, categories and concepts etc. are the few services that can be extracted with the help of NLU. Category extraction classifies the data into few categories with their confidence scores. Categories with confidence score less than 0.7 are eliminated in order to exclude those which are failing to conclude the tweets with low confidence scores. Following this filtration, overlap coefficient is calculated using equation 2 to identify overlapping categories between the consecutive windows.

$$OC(\text{Cat}(T_i, T_{i+1})) = \frac{|\text{Cat}(T_i) \cap \text{Cat}(T_{i+1})|}{\min(|\text{Cat}(T_i)|, |\text{Cat}(T_{i+1})|)} \quad (2)$$

The common categories are extracted between two intervals with two different confidence scores for the same category. This appears to be related to the problem

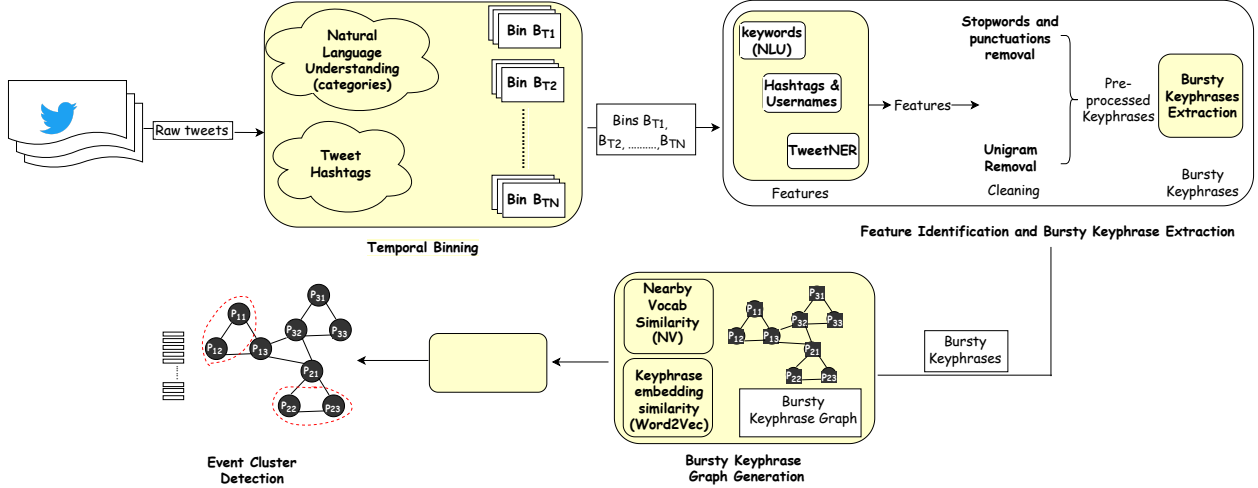


Fig. 1: Workflow of the proposed KEvent approach

of dynamic membership degrees. As a result, using fuzzy logic, the intersection of such a set is determined by selecting elements with minimum degrees of membership and common category score is calculated using equation 3. Thereafter, categorical convergence is calculated using equation 4, which integrates the scores from equations 2 and 3.

$$\begin{aligned}
 & CCS(Cat((T_i, T_{i+1}))) \\
 &= \frac{1}{L} \left(\sum_{K=1}^L (\min(\text{Score}(Cat_K(T_i)), \right. \\
 & \quad \left. \text{Score}(Cat_K(T_{i+1})))) \right) \quad (3)
 \end{aligned}$$

$$\begin{aligned}
 & CC(T_i, T_{i+1}) \\
 &= OC(Cat(T_i, T_{i+1})) \times CCS(Cat(T_i, T_{i+1})) \quad (4)
 \end{aligned}$$

3) **Mutual Convergence:** It is the linear combination of hashtag and categorical convergence scores and calculated with the help of equation 5. If mutual convergence in equation 5 is greater than user-defined threshold, θ , then the tweets are combined from both the intervals into a single bin with the instinct that the combined data contains semantically coherent tweets.

$$MC(T_i, T_{i+1}) = \alpha(HC(T_i, T_{i+1})) + \beta(CC(T_i, T_{i+1})) \quad (5)$$

B. Feature Identification and Bursty Keyphrase Extraction

When there is a presence of an event, a surge in tweets has always been noticed for that particular time. To this end, bursty keyphrases are one of the important indicators of the presence of event because there are

a few prominent keyphrases that users generally use while posting about the event. The insight behind bursty keyphrase extraction is the sudden spike in the frequency of some keyphrases in that moment that can indicate an actual event.

1) **Feature Identification:** After grouping tweets into a new set of intervals (bins), the next stage is to identify features from the tweets. A tweet is composed of a 280-characters long sequence of words and creativity of users to write a sentence in limited number of characters complicate the process to understand the sense of a sentence. As a result, this section aims to extract useful keyphrases from a tweet and to avoid noises. To this end, each tweet is divided into three parts – *keywords*, *hashtags* and *usernames/mentions*, and *NERtag* that are described in the following paragraphs.

- **Keywords:** The NLU service provided by IBM² is used to extract keywords, that help to represent a collection of tweets with valuable and contextually rich keyphrases.
- **Hashtags and usernames/mentions:** Hashtag is considered to be one of the major characteristics as it represents an event distinctively. Sometimes an event erupts with the hashtag as the dominant keyphrase. Another essential component is the username. Users propagate their tweets by referencing others. For example, when there is a political event, people communicate their voices to notable individuals linked with that event by mentioning them.
- **NERtag:** NER (named entity recognition) tagging transforms a tweet into meaningful keyphrases. As we endeavour to extract more relevant and representative terms from the limited text, we can retrieve

²<https://www.ibm.com/demos/live/natural-language-understanding/>

real-world things such as individuals, locations, and organisations among others. `Spacy`³, an excellent Python module, is used to extract entities.

2) **Bursty Keyphrase Extraction:** Among all the keyphrases obtained from the previous paragraph, a handful has burst within the period. Therefore, an bursty keyphrase extraction approach is adopted from [6] which is described in the following paragraph.

Let, N_B be the number of tweets in a bin B , and $f_{(KP,B)}$ is the frequency of tweets containing keyphrase KP in a bin B . Therefore, the probability of a keyphrase KP with frequency $f_{(KP,B)}$ can be considered as a Binomial Distribution $B(N_B, P_{KP})$, where, P_{KP} is the probability of observing keyphrase KP in any random bin. Since, number of tweets N_b can be large, this probability distribution can be estimated as a normal distribution with parameters $E[KP|B] = N_B P_{KP}$ and $\sigma[KP|B] = \sqrt{N_B P_{KP}(1 - P_{KP})}$.

As a result, if the keyphrase has frequency $f_{kp,b} \geq E[KP|B]$ which is expected mean of the keyphrase in current bin B , then that keyphrase is considered as a bursty keyphrase. Equation 6 is used to normalize the frequency of bursty keyphrase to the range (0,1), where $S(\cdot)$ is a sigmoid function, and since the sigmoid function level is well within the range [-10, 10], a constant 10 is inserted.

$$P_b(KP, B) = S\left(10 \frac{f_{KP,B} - (E[KP|B] + \sigma[KP|B])}{\sigma[KP|B]}\right) \quad (6)$$

[6] also integrated user frequency (number of different users tweeted a particular keyphrase in a tweet), retweet count of the keyphrase, and followers count of user. Therefore, the updated equation considering these facts is given in equation 7.

$$w_b(KP, B) = P_b(KP, B) \log(u_{KP,B}) \times \log(src_{KP,B}) \log(\log(sfc_{KP,B})) \quad (7)$$

Finally, using all of the keyphrases and their scores, the top- K keyphrases are chosen as bursty keyphrases according to their weight. $\sqrt{N_B}$ is chosen as an appropriate value of K .

C. Bursty Keyphrase Graph Generation and Event Detection

This subsection describes the processes of keyphrase graph generation and event detection. Further details about these functionalities are presented in the following sub-sections.

³<https://spacy.io>

1) **Bursty Keyphrase Graph Generation:** Bursty keyphrases of all the bins are combined and modeled as a weighted graph with multi-attributed edges. This graph is defined as $G(V, E)$, where V represents the set of vertices and $E \subseteq V \times V$ is the set of edges defining the relationship between the vertices. In this graph, vertices represent the bursty keyphrases and edges represent the semantic and lexical relationship between the vertices. For lexical similarity between the two vertices, overlap coefficient ($OC(N(V))$) of nearest vocab of two bursty keyphrases is calculated using equation 8. Thereafter, for each bursty keyphrase, the top five words based on the frequency of occurrence with the keyphrase is considered as a nearest vocab ($N(V)$) of the bursty keyphrase.

$$OC(NV(KP_a, KP_b)) = \frac{|NV(KP_a) \cap NV(KP_b)|}{\min(|NV(KP_a)|, |NV(KP_b)|)} \quad (8)$$

For contextual similarity, self-learned representation of keyphrases is utilized with the help of `Word2Vec`, where we trained the whole day dataset by tweaking the window parameter (W_t). An embedding vector (EV) is formed for each graph vertex. Intuition behind using `Word2Vec` is that keyphrases with similar context will have similar embedding representation [12]. Finally, Cosine similarity [13][14] is calculated between the embedding vectors of the keyphrases using equation 9.

$$CosineSim(EV(KP_a), EV(KP_b)) = \frac{EV(KP_a) \cdot EV(KP_b)}{|EV(KP_a)| \times |EV(KP_b)|} \quad (9)$$

Equation 10 is used to calculate a weighted linear combination of the similarities obtained from equations 8 and 9 for similarity graph generation.

$$CombinedSim(KP_a, KP_b) = \gamma(CosineSim(EV(KP_a), EV(KP_b))) + \delta(OC(NV(KP_a, KP_b))) \quad (10)$$

2) **Event Detection:** Markov clustering algorithm known as MCL is applied on the similarity graph to decompose it into different cohesive regions, each one representing a particular event. Markov clustering is a graph-based clustering that simulates a random walk on a graph which results in splitting the graph data into clusters [15]. In MCL, an inflation parameter denoted by r is responsible for strengthening and weakening current connections. For each cluster obtained from MCL, the top ten keyphrases are extracted as representative keyphrases using the bursty score calculated in the subsection III-B2. Clusters with less than three bursty keyphrases are deemed noise and deleted due to their no contribution in detecting important events.

IV. EXPERIMENTAL SETUP AND RESULTS

This section presents a brief description of dataset, parameters, ground truth for evaluation, and `KEvent` results followed by a comparative analysis.

A. Dataset and Parameters

Twitter data is drawn from the famous event detection benchmark dataset `EventCorpus2012` [7] for a month period. In `KEvent`, data for three days (from October 12 to October 14, 2012) is selected for experimentation to prove the efficacy of our proposed approach. The dataset contains around 5 million tweets for the mentioned period consisting of tweet texts with hashtags and mentions, user information, retweet count of a tweet, and follower count of the user. By default, the one-day data time window is divided into 24 hours. With the help of temporal binning, new time windows (bins) is defined for a day, where a user-defined threshold θ is mentioned. In our experiment, θ value is set to 0.5. In addition, a linear combination is employed in equation 5, where values of α and β are considered at 0.4 and 0.6, respectively. In subsection III-C1, window parameter (W_t) is set to 3. Further, combined similarity is calculated using equation 10, γ and δ are set to 0.25 and 0.75, respectively. Finally, the inflation parameter (r) in `MCL` is examined for various values and $r = 2$ is found as optimal one, as explained in section IV-C.

B. Ground Truth Description

Due to the unavailability of ground truth events in event detection dataset, previous studies considered events reported in news media and webpages as ground truth for a specified period. Seeking event details for Twitter data from webpages is not a great idea as news media and social media are two different platforms. Discussions on Twitter may vary depending on the users' interest. We can confirm Twitter's event occurrences with mainstream media reporting, but not the other way around. This situation results in a reduced recall. Our target is to find events in available Twitter data. As a result, for delineating the ground truth, the union of true positive events that correlate to real events by `KEvent` and the event list provided by `SEDTWIK` [6] is observed for above-mentioned dataset. All the events are manually checked with reliable news sources to authenticate the event information. Because it is difficult to accommodate all the events in this paper, the event list is published⁴ with all the news sources and categorization of an event into local and international events.

⁴<https://github.com/sielviesharma/KEvent>

C. Evaluation Metrics and Results

The proposed approach is evaluated using standard *precision*, *recall*, and *f-score* metrics. Using definitions provided by several researchers [3] [16], *precision* is calculated as distinct true positive events detected that correspond to real events to the total number of events detected, and *recall* is the number of all distinct true positive events to the events exist in the ground truth dataset. Finally, *F-score* is calculated as the harmonic mean of the *precision* and *recall*.

$$Precision = \frac{\#Correctly\ detected\ distinct\ events}{\#detected\ events} \quad (11)$$

$$Recall = \frac{\#Correctly\ detected\ distinct\ events}{\#ground\ truth\ events} \quad (12)$$

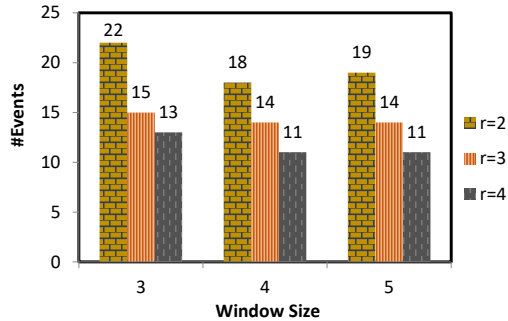
$$F-Score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \quad (13)$$

Total 62 events are retrieved across three days, as well as numerous major events missed by `SEDTWIK` and `Twevent`, are shown in the section IV-D. In addition, effects of various parameter settings on detection of important events is also studied. For `Word2Vec`, a window parameter is used to define the context of a word. Depending on the dataset, context of a word may vary with different window sizes. Therefore, after testing for several windows, optimal value is chosen that yields better experimental results. Experimentation has also been conducted using alternate techniques such as `fastText`, as it works on character-level learning and a word representation is a sum of all the character n-gram vectors. In terms of context, averaging `Word2Vec` vectors outperformed `fastText` with respect to social network data. In `MCL`, inflation parameter (r) influences model performance. Therefore, (r) is also evaluated for different setting in order to find a fitting value. Correlation of both variable parameters, `Word2Vec` window size, and inflation parameter of `MCL` with results is depicted in figure 2 and table II, respectively.

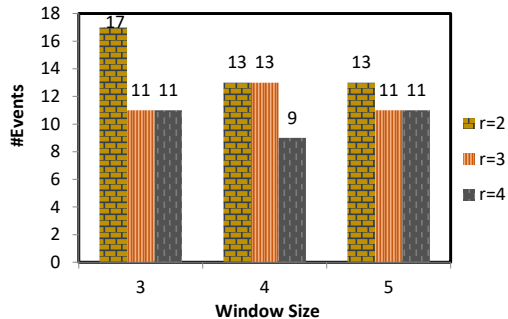
D. Comparative Analysis

In this study, we have considered the following state-of-the-art methods for event detection:

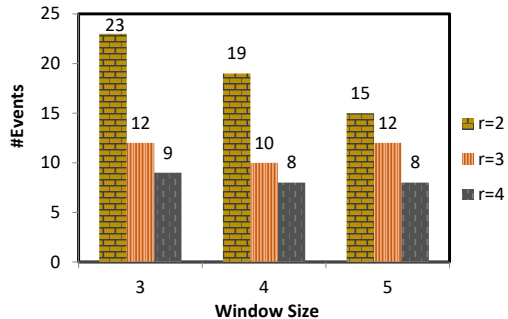
- **SEDTWIK[6]:** It is a segmentation-based method inspired by Wikipedia anchor titles where various Twitter features were used to extract bursty segments and cluster them to extract events.
- **Twevent[5]:** Similar to above, authors used segmentation-based event detection with the help of Microsoft N-gram and Wikipedia titles. Thereafter, bursty segments were identified based



(a)



(b)



(c)

Fig. 2: #Events detected with varying window size (W_t) and inflation parameter (r) for (2a) Oct 12, 2012, (2b) Oct 13, 2012 and (2c) Oct 14, 2012

on frequency patterns. Further, the Jarvis Patrick clustering algorithm was applied to extract events.

The effectiveness of our proposed approach is estimated in comparison to the above-mentioned methods. In Twevent, two scenarios are taken into consideration for segmentation. First scenario uses Wikipedia keyphrase-ness values $Q(s)^4$ instead of Microsoft-N gram because the service is no longer available. In second scenario, SEDTWik's segmentation method is applied because both the methods favor Wikipedia anchor titles while segmenting a tweet.

The comparison of KEvent with baselines is represented in tabular and graphical form in table IV and figure 3, respectively. It can be observed that our approach significantly outperforms the baselines, and many of the

TABLE II: #Events detected with varying window size (W_t) and inflation parameter (r) for (IIa) Oct 12, 2012, (IIb) Oct 13, 2012 and (IIc) Oct 14, 2012

Window Size	Inflation parameter (r)		
	2	3	4
3	22	15	13
4	18	14	11
5	19	14	11

(a)

Window Size	Inflation parameter (r)		
	2	3	4
3	17	11	11
4	13	13	9
5	13	11	11

(b)

Window Size	Inflation parameter (r)		
	2	3	4
3	23	12	09
4	19	10	08
5	15	12	08

(c)

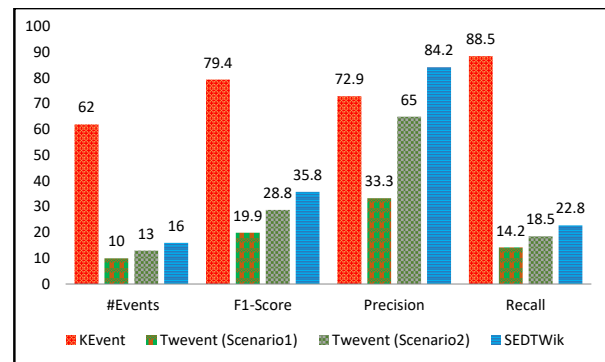


Fig. 3: Performance comparison of KEvent with SEDTWik and Twevent(Scenario 1 and 2)

key events listed in table III are missed by the baselines. In [17], the authors agreed that in addition to the events they have listed in previous work [7], many additional events can be derived from the same dataset. The event list for the same is not available; however, authors in [6] mentioned that their model, SEDTWik, extracted more number of events as compared to [7]. As our model is performing better than SEDTWik, we believe that our approach would also perform better than [7].

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed KEvent for event detection on Twitter data. The unsupervised methodology used by KEvent makes it bias-free. The KEvent is a graph-based semantic-enriched strategy to find potential events in tweets by extracting bursty keyphrases

TABLE III: Sample of events that are detected by only our proposed approach KEvent

Date	Event description	Validated via news sources
Oct 12, 2012	NASA SPACE SHUTTLE ENDEVOUR ARRIVES AT CALIFORNIA SCIENCE CENTRE AFTER 2- DAY TRIP THROUGH LOS ANGELES	CNN, ECONOMIC TIMES, NASA.GOV
	TURKEY SCRAMBLES FIGHTER PLANES TO SYRIA BORDER	CNN, BBC, REUTERS
Oct 13, 2012	SHOTS FIRED AT OBAMA CAMPAIGN OFFICE DENVER	REUTERS, THE HINDU
	SYDNEY FC Vs NEW CASTLE SOCCER MATCH	ESPN, SKYSPORTS
Oct 14, 2012	HEATHER WATSON WINS JAPAN OPEN AND WTA TITTLE	BBC, ESPN
	CHENNAI SUPER KING VS SYDNEY SIXERS	ESPN, ICC CRICKET, TIMES OF INDIA

TABLE IV: Performance comparison of KEvent with SEDTWik and Twevent (Scenario 1 and 2)

Approach	#Events	Precision	Recall	F-Score
KEvent	62	72.9	88.5	79.4
SEDTWik [6]	16	84.2	22.8	35.8
Twevent (Scenario1) [5]	10	33.3	14.2	19.9
Twevent (Scenario 2) [5]	13	65	18.5	28.8

from the data and building a weighted graph of those keyphrases with self-learned contextual representation to deal with the arbitrariness of the data. Markov clustering is also used to identify key events. In comparison to state-of-the-art approaches, adding semantics to other existing features improved our model’s performance and significantly increased the number of events that were recognized. Future work on event detection in OSN may include investigating and leveraging the dependence and relationships between two events. The connection between two occurrences can be used to describe or better understand how an event developed. If numerous events are connected to a single event, we can improve the event detection process by looking into the elements that drive splitting and merging.

REFERENCES

[1] F. Atefeh and W. Khreich, “A survey of techniques for event detection in twitter,” *Computational Intelligence*, vol. 31, no. 1, pp. 132–164, 2015.

[2] N. Panagiotou, I. Katakis, and D. Gunopulos, “Detecting events in online social networks: Definitions, trends and challenges,” *Solving Large Scale Learning Tasks: Challenges and Algorithms*, pp. 42–84, 2016.

[3] J. Weng and B.-S. Lee, “Event detection in twitter,” in *Proceedings of the Int’l AAAI Conference on Web and Social Media*, vol. 5, no. 1, 2011, pp. 401–408.

[4] H. Kwak, C. Lee, H. Park, and S. Moon, “What is twitter, a social network or a news media?” in *Proceedings of the 19th Int’l Conference on World Wide Web*, Apr 2010, pp. 591–600.

[5] C. Li, A. Sun, and A. Datta, “Twevent: Segment-based event detection from tweets,” in *Proceedings of the 21st ACM Int’l Conference on Information and Knowledge Management*, Oct 2012, pp. 155–164.

[6] K. Morabia, N. L. B. Murthy, A. Malapati, and S. Samant, “SEDTWik: Segmentation-based event detection from tweets using wikipedia,” in *Proceedings of the Conference of the North American Chapter of the ACL: Student Research Workshop*, June 2019, pp. 77–85.

[7] A. M. J, Y. Moshfeghi, and J. M. Jose, “Building a large-scale corpus for evaluating event detection on twitter,” in *Proceedings of the 22nd ACM Int’l Conference on Information and Knowledge Management*, Oct 2013, pp. 409–418.

[8] Q. Li, A. Nourbakhsh, S. Shah, and X. Liu, “Real-time novel event detection from social media,” in *IEEE 33rd Int’l Conference on Data Engineering*, Apr 2017, pp. 1129–1139.

[9] H. Becker, M. Naaman, and L. Gravano, “Beyond trending topics: Real-world event identification on twitter,” in *Proceedings of the Int’l AAAI Conference on Web and social media*, vol. 5, no. 1, Aug 2011, pp. 438–441.

[10] H. Hettiarachchi, M. Adedoyin-Olowe, J. Bhogal, and M. M. Gaber, “Embed2Detect: Temporally clustered embedded words for event detection in social media,” *Machine Learning*, vol. 111, no. 1, pp. 49–87, 2022.

[11] X. Zhou and L. Chen, “Event detection over twitter social media streams,” *The VLDB journal*, vol. 23, no. 3, pp. 381–400, 2014.

[12] C. Comito, A. Forestiero, and C. Pizzuti, “Word embedding based clustering to detect topics in social media,” in *IEEE/WIC/ACM Int’l Conference on Web Intelligence*, Oct 2019, pp. 192–199.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[14] M. Antoniak and D. Mimno, “Evaluating the stability of embedding-based word similarities,” *Transactions of the ACL*, vol. 6, pp. 107–119, 2018.

[15] S. V. Dongen, “Graph clustering via a discrete uncoupling process,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 121–141, Feb 2008.

[16] J. G. C. James Allan, G. Doddington, J. Yamron, and Y. Yang, “Topic detection and tracking pilot study final report,” 1998.

[17] A. M. J and J. M. Jose, “Real-time entity-based event detection for twitter,” in *Int’l Conference of the Cross-Language Evaluation Forum for European Languages*, Nov 2015, pp. 65–77.