SEDTWik: Segmentation-based Event Detection from Tweets using Wikipedia

Keval M. Morabia, Neti Lalita Bhanu Murthy, Aruna Malapati and **Surender S. Samant**
Birla Institute of Technology and Science (BITS), Pilani
Hyderabad, India
kevalmorabia97@gmail.com,
{bhanu, arunam, surender.samant}@hyderabad.bits-pilani.ac.in

Abstract

Event Detection has been one of the research areas in Text Mining that has attracted attention during this decade due to the widespread availability of social media data specifically twitter data. Twitter has become a major source for information about real-world events because of the use of hashtags and the small word limit of Twitter that ensures concise presentation of events. Previous works on event detection from tweets are either applicable to detect localized events or breaking news only or miss out on many important events. This paper presents the problems associated with event detection from tweets and a tweet-segmentation based system for event detection called SEDTWik, an extension to a previous work, that is able to detect newsworthy events occurring at different locations of the world from a wide range of categories. The main idea is to split each tweet and hash-tag into segments, extract bursty segments, cluster them, and summarize them. We evaluated our results on the well-known Events2012 corpus and achieved state-of-the-art results.

Keywords: Event detection, Twitter, Social Media, Microblogging, Tweet segmentation, Text Mining, Wikipedia, Hashtag.

1 Introduction

Microblogging, as a form of social media, is fast emerging in this decade. One of the best examples for this is Twitter which allows 280-character limit for a tweet. It is used not only to share and communicate with friends and family but also as a medium to share real-world events. An *event* according to the Topic Detection and Tracking (TDT) project (Allan et al., 1998), is "some unique thing that happens at some point in time". Becker et al. (2011) defines an event as "a real-world occurrence e with an associated time period T_e and a time-ordered stream of Twitter messages M_e, of substantial volume, discussing the occurrence and published during time T_e". We borrow these definitions of an event in our work.

In Twitter, a user can not only publish about an event but can also propagate by *retweeting* the post by someone else. A user can also attach a *hashtag* with the tweet which can provide a significant amount of information about the event (e.g., **#RIP** to signify that the tweet is related to someone's death). But some hashtags can also be used to promote ideas known as *memes* (Kotsakos et al., 2014). Event detection from tweets also faces other challenges like noisy data, informal writing, grammatical errors, and a large volume of data coming at very high velocity. According to Internet Live Stats[1], on an average 6,000 tweets are published every second, which corresponds to nearly 500 million tweets per day. Moreover, nearly 40% of these tweets are just "pointless babbles"[2] which are insignificant to the task of event detection.

To tackle the above-mentioned challenges, we present SEDTWik - a tweet segmentation-based event detection system that utilizes an external knowledge base like Wikipedia. The rest of the paper is organized as follows. Section 2 describes the working of SEDTWik in detail. Section 3 presents our experimental results. Section 4 presents some related works in event detection. We conclude in section 5 along with future work to be done.

2 SEDTWik

In this section, we present SEDTWik, an extension of a previous work by Li et al. (2012a) called Twevent. SEDTWik is an event detection

[1] http://www.internetlivestats.com/twitter-statistics

[2] https://pearanalytics.com/blog/2009/twitter-study-reveals-interesting-results-40-percent-pointless-babble

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 77–85
Minneapolis, Minnesota, June 3 - 5, 2019.

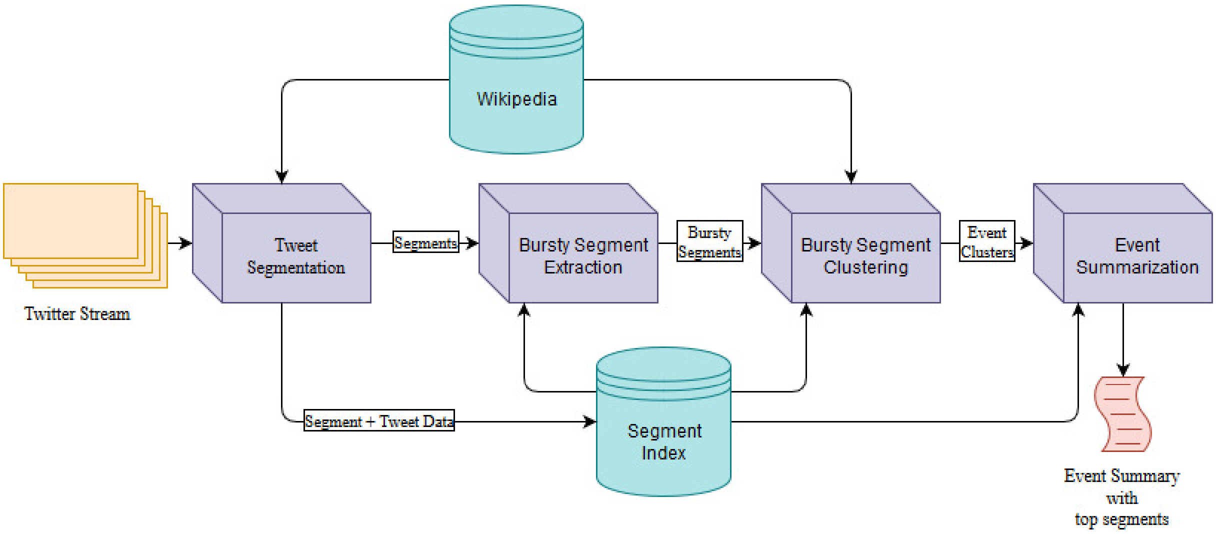

Figure 1: SEDTWik Architecture.

framework that consists of four components: *tweet segmentation, bursty segment extraction, bursty segment clustering,* and *event summarization.* Figure 1 shows the architecture of SEDTWik. Events are detected from a time window t of a fixed length during which all the tweets published are processed. In the tweet segmentation phase, all tweets coming from the Twitter stream within the current time window are segmented, and the segments along with the tweet details are indexed for use in next stages. Hashtags are given more weight as they contain more information. Based on the probability distribution of segments, retweet counts, user diversity, and user popularity, abnormally bursty segments are extracted and clustered in the next two stages. Finally, the clusters are summarized in the last step. In the rest of this section, we present all the four components in detail.

2.1 Tweet Segmentation

Tweet segmentation was introduced by Li et al. (2012b) and Li et al. (2015) for Named Entity Recognition (NER) in which they used a dynamic programming based approach to segment tweets based on a "stickiness" score of a segment. In this section, we present an alternative approach using Wikipedia Page Titles Dataset[3] for segmentation of tweets and hashtags.

The task of tweet segmentation is to split a given tweet into non-overlapping meaningful segments. A segment can be unigram (a word) or multi-gram (a phrase). The reason why tweet segmentation is used is that a phrase contains much more specific information than the unigrams in it. So, a tweet segment makes the event more interpretable. For example, **[vice presidential debate]** is much more informative then **[vice], [presidential],** and **[debate]** separately that might be in any random order. While segmenting a tweet, we emphasize three components: *tweet text, name mentions,* and *hashtags*.

We consider *tweet text* as everything a user writes in a tweet except URL links, hashtags, and name mention. From tweet text, we only keep those segments that are present as a title of a Wikipedia page[3]. This ensures that only named entities (e.g., Barack Obama) or meaningful segments (e.g., new music) are kept from tweet text, and unnecessary words are removed that would otherwise increase noise in the event detection process.

Most Twitter users use a *name mention* in a tweet to mention a person by their username (e.g., **@iamsrk** for **Shah Rukh Khan**). So, we replace the username by their actual name and consider it as a segment.

The most important component in out event detection model is *hashtags*. Hashtags contain a lot of information in a concise form, and related tweets generally contain the same hashtag. Ozdikis et al. (2012a) used only hashtags for event detection as contrasted with Ozdikis et al. (2012b) and were able to get better results in the former. This motivated us to give more weight to hashtags in the segmentation process.

[3] http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-all-titles-in-ns0.gz

We use $\mathcal{H}$ as *hashtag weight*, so an $\mathcal{H}$ value of 2 means that all hashtags are duplicated in the segmentation process, resulting in a twice weight that would allow hashtags to become more bursty in the next stage. This also ensures that if a segment is not previously seen in the Wikipedia page titles, then its use in hashtag would still make the segment bursty. Since hashtags do not contain whitespace or punctuations, we consider the capitalization of letters to segment a hashtag. For example, **#BreakingNews** will be segmented as **[breaking news]**. Hashtags that do not contain any capitalization of letters in them would be considered as a unigram when segmenting them.

2.2 Bursty Segment Extraction

Since there are hundreds of thousands of unique segments within a day, clustering all of them for detecting events would be a computationally expensive task. So, once the tweets are segmented, we find out abnormally bursty segments that might be related to an event and discard the remaining ones.

Let N_t denote the number of tweets within the current time window t and $f_{s,t}$ be the number of tweets containing segment s in t. The probability of observing s with a frequency $f_{s,t}$ can be considered as a Binomial distribution $B(N_t, p_s)$ where p_s is the expected probability of observing segment s in any random time window. Since N_t is very large in case of tweets, this probability distribution can be approximated to a Normal distribution with parameters $E[s|t] = N_t p_s$ and $\sigma[s|t] = \sqrt{N_t p_s(1 - p_s)}$.

If a segment has $f_{s,t} >= E[s|t]$, it will be called a bursty segment, while a segment with $f_{s,t} < E[s|t]$ will not be considered bursty and will be discarded. We use a formula for the *bursty probability* $P_b(s,t)$ for segment s in time window t defined by Li et al. (2012a) as given in (1) that transfers the frequency of a bursty segment to the range (0,1).

$$P_b(s,t) = S(10\frac{f_{s,t} - (E[s|t] + \sigma[s|t])}{\sigma[s|t]}) \quad (1)$$

where $S(\bullet)$ is the sigmoid function, and since sigmoid function smooths well in the range [-10,10], the constant 10 is introduced.

Instead of depending entirely on tweet frequency, to incorporate user diversity, *user frequency* $u_{s,t}$ is also used which denotes the number of distinct users using segment s in time window t. A *retweet* is a copy of a tweet created by another user. According to Boyd et al. (2010), a retweet is “a conversational practice and can negotiate authorship, attribution, and communicative fidelity”. We find that a tweet retweeted by many users might be related to an important event and can be used to provide more weight to segments in retweets. We define *segment retweet count* of a segment s in t as $src_{s,t}$ which is the sum of retweet counts of all tweets containing s in t. A tweet by someone who has millions of followers (e.g., a celebrity or a news page) might also be more important as compared with someone who has very few followers. Giving more weight to such tweets will ensure that spam or self-promoting tweets are filtered out and do not harm the accuracy of the event detection process. So, we define *segment follower count* of a segment s in t as $sfc_{s,t}$ which is the sum of follower count of all users using this segment in t. Combining all the above, the formula for *bursty weight* $w_b(s,t)$ for segment s in t is defined in (2).

$$\begin{aligned} w_b(s,t) = P_b(s,t)log(u_{s,t}) \times \\ log(src_{s,t})log(log(sfc_{s,t})) \end{aligned} \quad (2)$$

Among all the segments, top $\mathcal{K}$ segments are selected as bursty segments based on their bursty weight. A small value of $\mathcal{K}$ would result in a very low recall of events detected, and a large value of $\mathcal{K}$ may bring in more noise leading to higher computational cost. Therefore, an optimal value of $\mathcal{K}$ is kept to be $\sqrt{N_t}$.

2.3 Bursty Segment Clustering

In this section, we cluster bursty segments and filter non-event clusters using the approach by Li et al. (2012a).

Since the topics in tweets are fast changing and extremely dynamic, the similarity of two segments is calculated from their temporal frequency and the contents of the tweets that contain the segment. Each time window is evenly split into $\mathcal{M}$ subwindows $t =< t_1, t_2, ..., t_M >$. Let $f_t(s,m)$ be the tweet frequency of segment s in the subwindow t_m and $T_t(s,m)$ be the concatenation of all the tweets in the subwindow t_m that contain segment s. The similarity $sim_t(s_a, s_b)$ between segments s_a and s_b in time window t is calculated based on formula (3).

$$sim_t(s_a, s_b) = \sum_{m=1}^{M} w_t(s_a, m) w_t(s_b, m) \times \\ sim(T_t(s_a, m), T_t(s_b, m)) \tag{3}$$

where $w_t(s, m)$ is the fraction of frequency of segment s in the subwindow t_m as mentioned in (4) and $sim(T_1, T_2)$ is the tf-idf similarity of the set of tweets T_1 and T_2.

$$w_t(s, m) = \frac{f_t(s, m)}{f_{s,t}} \tag{4}$$

Using the similarity measure given in (3), all the bursty segments are clustered using a variation of Jarvis-Patrick algorithm (Jarvis and Patrick, 1973). In this, all segments are considered as nodes and initially, all nodes are disconnected. An edge is added between segments s_a and s_b if k-Nearest neighbors of s_a contains s_b and vice versa. After adding all possible edges, all the connected components of the graph are considered as candidate event clusters. Those segments that do not have any edges are discarded from further processing.

After clustering the bursty segments, we found that some clusters were not related to any event. For example, one of the candidate event clusters detected from tweets of Sunday, October 14, 2012 had segments like **[sunday dinner], [sunday night], [every sunday], [sunday funday],** and **[next sunday]**. This kind of events have segments that are bursty on specific days of the week. Thus, some filtering has to be done to eliminate these events. So, use of external knowledge base like Wikipedia is made.

The newsworthiness $\mu(s)$ of a segment s, is defined as given in (5) which ensures that if a sub-phrase of a segment is an important phrase then the segment is also considered newsworthy.

$$\mu(s) = \begin{cases} e^{Q(s)} & \text{s is a word} \\ \max_{l \epsilon s} e^{Q(l)} - 1 & \text{otherwise} \end{cases} \tag{5}$$

where l is any sub-phrase of segment s and $Q(l)$ is the probability of l appearing as anchor text in Wikipedia articles containing l.

The newsworthiness $\mu(e)$ of an event cluster e, is defined in (6) that considers the newsworthiness of its constituent segments and the weight of edges of the event cluster in the form of segment similarity.

$$\mu(e) = \frac{\sum_{s\epsilon e_s} \mu(s)}{|e_s|} \frac{\sum_{g\epsilon E_e} sim(g)}{|e_s|} \tag{6}$$

where e_s is the set of segments associated with event e, E_e is the set of edges between segments of the event e, and $sim(g)$ is the similarity between nodes of the edge g which is calculated from (3).

Candidate events that are not likely to be realistic events are observed to have very small newsworthiness as compared to real events. So, if an event e satisfies the condition $\frac{\mu_{max}}{\mu(e)} < \mathcal{T}$ then only it is kept as a realistic event otherwise discarded. Here μ_{max} is the highest newsworthiness among all candidate event clusters and $\mathcal{T}$ is a threshold.

2.4 Event Summarization

A list of segments associated with an event cluster might not provide all the information related to an event. So, we used the LexRank algorithm (Erkan and Radev, 2004) to summarize the event clusters obtained in the previous step. The LexRank algorithm takes as input multiple documents and provides a summary of it by combining the top-ranking sentences. To summarize an event, we use all the tweets in current time window t that contain the segments in the event cluster obtained from the segment index created in the tweet segmentation phase and apply the algorithm to provide a summary of the event.

3 Experimental Results

In this section, we will mention the dataset and evaluation metrics we used, the statistics about tweet segmentation, and our results. Our model outperforms Twevent (Li et al., 2012a) with better precision, a greater number of events, and less duplicate events.

3.1 Dataset and Experimental Setting

The Wikipedia page titles dataset used in subsection 2.1 was a dump from March 2018 which contains 8,007,358 page titles. We used the Wikipedia keyphraseness values $Q(s)$[4] used by Li et al. (2012a) which was based on a dump released on Jan 30, 2010, and contains 4,342,732 distinct entities that appeared as anchor text.

[4] https://www.ntu.edu.sg/home/axsun/datasets.html

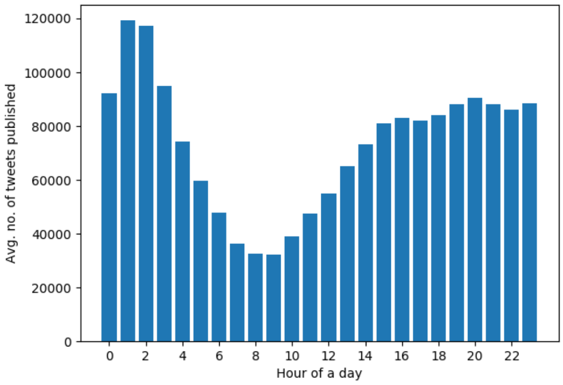

Figure 2: Tweet volume vs hour of day.

McMinn et al. (2013) created a Twitter corpus called Events2012 containing tweets from Oct 10 - Nov 7, 2012. They removed tweets containing more than 3 hashtags, 3 name mentions, or 2 URLs as they might be spam (Benevenuto et al., 2010). After all this filtering, the corpus contains over 120 million tweets. It also contains a list of 506 events detected in the corpus distributed among 8 categories. We used this corpus to estimate the segment probabilities p_s used in subsection 2.2 and to evaluate the performance of our model. Both Wikipedia Page Titles and the tweets in the corpus were preprocessed using using pyTweetCleaner[5]. Figure 2 shows a plot of average no. of tweets published within each hour of the day for this corpus.

There were several parameters that affect the performance of our model like time window size, number of subwindows $\mathcal{M}$, hashtag weight $\mathcal{H}$, number of neighbors k while clustering, and threshold $\mathcal{T}$. We set a time window to be of 24 hours which contains $\mathcal{M}$ = 12 subwindows of 2 hours each. We set $\mathcal{H}$ = 3, k = 3 neighbors and $\mathcal{T}$ = 4 in our work.

Allan et al. (1998) define **precision** as "the fraction of the detected events that are related to a realistic event". Moreover, Li et al. (2012a) defines another measure called **Duplicate Event Rate (DERate)** as "the percentage of events that have been duplicately detected among all realistic events detected". We use these definitions of *precision* and *DERate* in our evaluation. We did not use **recall** as a measure to evaluate the results found by our model because we find a lack of an exhaustive list of events in the Events2012 dataset (McMinn et al., 2013). Although they have provided a list of 506 events detected by their model within the period of Oct 10 - Nov 7, 2012, our model SEDTWik finds 48 events within a period of Oct 11 - Oct 17, 2012, that were not reported by them. Their later work (McMinn and Jose, 2015) also agrees with this. Table 1 shows some of the events detected by SEDTWik that were not detected by McMinn et al. (2013). Note that the event info is manually written since the summary generated is a set of tweets that is quite large to fit in the table. Instead of *recall*, we use **No. of events**, which is the number of realistic events detected, as a measure to evaluate the performance of SEDTWik.

Date	**Event Info**
Oct 11	International Day of the Girl Child
Oct 12	Justin Bieber and Nicki Minaj's music video of Beauty and a beat released
Oct 13	National No Bra Day
Oct 14	Korean Grand Prix F-1 racing in which Sebastian Vettel won for the third consecutive year
Oct 15	Little Nemo in Slumberland by Winsor McCay anniversary (released on this day in 1905)
Oct 16	The Great British Bake Off 2012 finals
Oct 17	A live episode of the UK soap opera Emmerdale was broadcast, marking its 40th anniversary

Table 1: Some events not detected by McMinn et al. (2013) that were detected by SEDTWik during the period of Oct 11 - Oct 17, 2012.

3.2 Tweet Segmentation Statistics

We segmented tweets from Oct 11 - Oct 17, 2012. After removing all the retweets, this period contained 11,705,978 tweets containing 3,653,039 distinct segments. Figure 3 shows the length of the segment along with their frequency within this period. We found that many of the bigrams were named entities (e.g., **[nicki minaj]**, **[mitt romney]**) or meaningful segments (e.g., **passed away**). Sample tweet segmentations with hashtag weight $\mathcal{H}$ = 3 are shown in Table 2. Notice that in the second row, the username **@ddlovato** corresponds to **Demi Lovato**, a pop singer. Thus,

[5] https://github.com/kevalmorabia97/pyTweetCleaner

Tweet	**Segmentation**
Joe Biden and Paul Ryan will be seated at the debate tonight **#VpDebate**	[joe biden], [paul ryan], [seated], [debate], [tonight], **[vp debate]**x3
My **#TeenChoice** for **#ChoiceSnapchatter** is *@ddlovato*	**[teen choice]**x3, **[choice snapchatter]**x3, *[demi lovato]*
Amanda Todd took her own life due to cyber bullying **#RipAmandaTodd** **#NoMoreBullying**	[amanda todd], [cyber bullying], **[rip amanda todd]**x3, **[no more bullying]**x3

Table 2: Sample tweet segmentations with $\mathcal{H} = 3$. Note that "x3" in the segmentation column signifies that the segment is present 3 times in the segmentation.

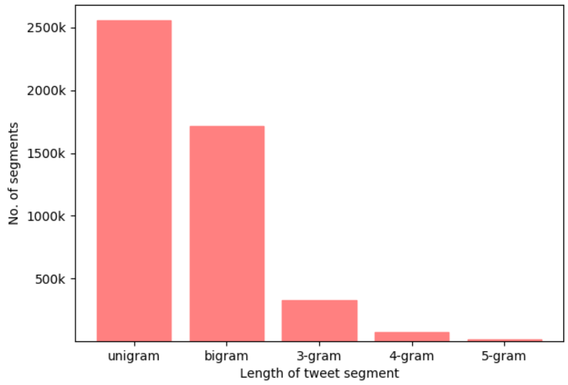

Figure 3: Segment length distribution during the period of Oct 11 - Oct 17, 2012.

Method	**No. of events**	**Precision**	**DERate**
SEDTWik	**79**	**88.12%**	**14.10%**
Twevent	42	80.32%	16.67%

Table 3: Comparison of SEDTWik (our method) with Twevent for events detected during the period of Oct 11 - Oct 17, 2012.

replacing username with the actual name makes the segment more interpretable.

3.3 Event Detection Results

Twevent (Li et al., 2012a) performs event detection from tweets using tweet segmentation and outperformed EDCoW (Weng and Lee, 2011) which was the state-of-the-art method that time, in terms of more no. of events, higher precision and recall, and less duplication rate. Since our model SEDTWik is an extension of Twevent, we set Twevent's results as a baseline for our model and compare both these models in this section.

Table 3 shows the comparison of SEDTWik with Twevent in terms of *no. of events, precision,* and *DERate* for events detected in the period of Oct 11 - Oct 17, 2012. Recall that we are not calculating *recall* of our model because of lack of an exhaustive list of events within this period. Note that for calculating the results of Twevent, instead of Microsoft Web N-gram service, we used our estimates of probability which was also used in our model for the same task. The reason is that Microsoft has discontinued providing this Web-service. After the event clusters and summary are generated, we manually annotate the clusters as realistic or non-realistic event and calculate the precision on them.

As shown in Table 3, SEDTWik achieved a *precision* of 88.12% as compared to 80.32% by Twevent. The *no. of events* detected by SEDTWik were significantly more than that by Twevent (79 vs 42). In terms of *DERate*, SEDTWik performs slightly better than Twevent (14.10% vs 16.67%). Thus, our model SEDTWik outperforms Twevent in all the three metrics.

Edouard et al. (2017) and TwitterNews+ (Hasan et al., 2016) also evaluated their models on the same Events2012 dataset (McMinn et al., 2013) but on a different period of tweets and were able to get precision values of 75.0% and 78.0% only. Since we have to manually annotate the results, we did not re-evaluate the results of our model on these tweets but we believe our model would outperform both these models in terms of precision.

Table 4 shows some of the events detected by SEDTWik for each day in the period Oct 11 - Oct 17, 2012, along with top segments in the event cluster. Note that the event information is manually written since the summary consists of several tweets that is quite large to fit in the table.

SEDTWik code, the data used, and the entire

Date	Event
Oct 11	• **[mo yan], [chinese writer], [nobel prize literature]** → Chinese author Mo Yan wins the Nobel Prize in Literature. • **[national coming out day], [national coming day], [lgbt], [coming day], [ncod]** → National Coming Out Day celebrated on this day. • **[steelers], [nfl], [titans], [tnf]** → Pittsburgh Steelers vs. Tennessee Titans Thursday Night Football (TNF) game.
Oct 12	• **[nobel peace prize], [nobel], [european union], [peace prize]** → The European Union wins the 2012 Nobel Peace Prize. • **[nlds], [st louis cardinals], [cardinal nation], [washington nationals]** → St. Louis Cardinals win their National League Divisional Series (NLDS) against Washington Nationals.
Oct 13	• **[xfactor], [x factor], [james arthur], [rylan clark]** → X Factor UK finalists James Arthur and Rylan Clark give a live show in London. • **[national no bra day], [no bra day], [th october]** → National No Bra Day celebrated on 13th October.
Oct 14	• **[arlen specter], [passed away], [sen arlen specter]** → Former US Senator Arlen Specter, died at the age of 82. • **[taylor swift], [xfactor], [the x factor]** → Pop singer Taylor Swift performs live at the X Factor UK.
Oct 15	• **[justin bieber], [baabworldrecord], [vevo]** → Justin Bieber's music video Beauty and a Beat (BAAB) creates world record of most watched VEVO video in 24 hrs. • **[breast cancer awareness month], [breast cancer awareness], [cure cancer]** → Every year, October is celebrated as Breast Cancer Awareness Month.
Oct 16	• **[debate], [barack obama], [presidential debate]** → 2nd US presidential debate between Barack Obama and Mitt Romney. • **[hilary mantel], [man booker prize], [booker prize]** → Hilary Mantel wins the 2012 Man Booker Prize for her novel.
Oct 17	• **[lance armstrong], [endorsement deal], [nike]** → Nike ended the promotional agreements they had with Lance Armstrong when he was accused of using performance enhancing drugs. • **[emmerdale live], [emmerdalelive], [live love]** → A live episode of the UK soap opera Emmerdale was broadcast, marking its 40th anniversary.

Table 4: Some of the events detected by SEDTWik for each day in the period Oct 11 - Oct 17, 2012, along with top segments in the event cluster.

list of events detected can be found here[6].

3.4 Impact of $\mathcal{H}$ and $\mathcal{T}$

Recall that while performing tweet segmentation in subsection 2.1, we used $\mathcal{H}$ as hashtag weight which signifies by how many times, the frequency of a hashtag is multiplied. As most users associate a hashtag with any important tweet and that hashtag is common among similar tweets, giving more weight to hashtags seems intuitive. A lower value of $\mathcal{H}$ would cause noisy segments from tweet text to dominate and harm the accuracy of the event detection model. Similarly, a higher value of $\mathcal{H}$ would not allow other frequently used segments in the tweet text to become bursty and again reduce the accuracy. We experimented with $\mathcal{H}$ values 1,2,3, and 4, and found the best results at $\mathcal{H} = 3$.

The threshold $\mathcal{T}$ was used in deciding if a candidate event cluster is a realistic event or not in subsection 2.3. We observed that on increasing $\mathcal{T}$, more event would be considered realistic that would increase the number of events detected, but reduce the precision of the model. On experimenting with different values of $\mathcal{T}$ from 2,3,4, and 5, we found optimal results at $\mathcal{T} = 4$.

[6] https://github.com/kevalmorabia97/SEDTWik-Event-Detection-from-Tweets

4 Related Work

Event detection from tweets is not a new topic of research, but rather an area on which extensive research has been done over this decade. In this section, we will present some of the related works in event detection from tweets that have motivated us to research in this field.

Panagiotou et al. (2016), Weiler et al. (2016) and Farzindar and Khreich (2015) presented a survey of many approaches that have been used over the past few years for event detection from Twitter. They had also mentioned several open challenges for event detection from tweets. Our work has tried to address some of these challenges in this paper.

TwInsight (Valkanas and Gunopulos, 2013) identified events by monitoring surges in 6 emotional states (anger, fear, disgust, happiness, sadness, and surprise) and gave information about the location, timestamp, emotion, and description of the event.

EvenTweet (Abdelhaq et al., 2013) used a fixed historical usage of words for finding those that have a burstiness degree two standard deviation above mean and then clustered them. Their method was used to find localized events only. EventRadar (Boettcher and Lee, 2012) also used Twitter to detect local events like parties and art exhibitions.

McMinn et al. (2013) created a pool of events using Locality Sensitive Hashing (LSH), Cluster Summarization, and Wikipedia, and clustered them using category, temporal, and content-based features and found events distributed among 8 categories (e.g., Sports, Science & Technology, etc.). But as shown in Table 1, there were many events that were not detected by their model.

Phuvipadawat and Murata (2010) used features like hashtags, usernames, follower count, retweet count, and proper noun terms to cluster and rank breaking news detected from Twitter.

Twevent (Li et al., 2012a) used a segmentation-based event detection from tweets method. In this method, tweets were segmented using a "stickiness" score of segments, and bursty segments were selected based on the prior probability distribution of segments, and user diversity, and were clustered into events. Our work SEDTWik is extension of Twevent so we have compared these two methods in subsection 3.3.

Recently, ArmaTweet (Tonon et al., 2017) used semantic event detection on Tweets to detect events such as 'politician dying' and 'militia terror act'.

5 Conclusion And Future Work

Twitter has experienced an explosive increase in both users and the volume of information in the recent time which has attracted great interests from both industry and academia. Tweets being short and containing noisy data in large volume poses challenges on event detection task. In this paper, we presented SEDTWik - a tweet segmentation-based event detection system in which hashtags, retweet count, user popularity, and follower count were key features. Giving more weight to hashtags significantly improved the model's performance. Our model achieved outstanding results on event detection from tweets using Wikipedia. As part of future work, we will explore ways to improve the segmentation process and use URL links in the event detection process. We will try to find more efficient ways to estimate segment probabilities considering the days and months in which specific segments are observed. We also plan to apply more sophisticated methods for event summarization that also leverage the segment and cluster information instead of just using the them to find tweets to use for summarization.

Acknowledgments

We thank Dr. Aixin Sun, Associate Professor at Nanyang Technological University (NTU), Singapore for providing the Wikipedia keyphraseness values Q(s) that was used in calculating segment newsworthiness in subsection 2.3. We would also like to thank Mr. Siddhant Shenoy and Mr. Nitish Ravishankar for providing their valuable feedbacks on our earlier drafts.

References

Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. 2013. Eventweet: Online localized event detection from twitter. *Proc. VLDB Endow.*, 6(12):1326–1329.

James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, et al. 1998. Topic detection and tracking pilot study: Final report.

Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*.

Fabrcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virglio Almeida. 2010. Detecting spammers on twitter. In *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*.

Alexander Boettcher and Dongman Lee. 2012. Eventradar: A real-time local event detection scheme using twitter stream. In *2012 IEEE International Conference on Green Computing and Communications*, pages 358–367.

Danah Boyd, Scott Golder, and Gilad Lotan. 2010. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10.

Amosse Edouard, Elena Cabrio, Sara Tonelli, and Nhan Le Thanh. 2017. Graph-based event extraction from twitter. In *RANLP*.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Atefeh Farzindar and Wael Khreich. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31:132–164.

Mahmud Hasan, Mehmet A. Orgun, and Rolf Schwitter. 2016. Twitternews+: A framework for real time event detection from the twitter data stream. In *Social Informatics*, pages 224–239, Cham. Springer International Publishing.

Raymond A. Jarvis and Edward A. Patrick. 1973. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, C-22(11):1025–1034.

Dimitrios Kotsakos, Panos Sakkos, Ioannis Katakis, and Dimitrios Gunopulos. 2014. #tag: Meme or event? In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM ’14, pages 391–394, Piscataway, NJ, USA. IEEE Press.

Chenliang Li, Aixin Sun, and Anwitaman Datta. 2012a. Twevent: Segment-based event detection from tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM ’12, pages 155–164, New York, NY, USA. ACM.

Chenliang Li, Aixin Sun, Jianshu Weng, and Qi He. 2015. Tweet segmentation and its application to named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 27:558–570.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012b. Twiner: Named entity recognition in targeted twitter stream. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’12, pages 721–730, New York, NY, USA. ACM.

Andrew J. McMinn, Yashar Moshfeghi, and Joemon M. Jose. 2013. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM ’13, pages 409–418, New York, NY, USA. ACM.

Andrew James McMinn and Joemon M. Jose. 2015. Real-time entity-based event detection for twitter.

Ozer Ozdikis, Pinar Senkul, and Halit Oguztuzun. 2012a. Semantic expansion of hashtags for enhanced event detection in twitter. In *Proceedings of the 1st International Workshop on Online Social Systems*.

Ozer Ozdikis, Pinar Senkul, and Halit Oguztuzun. 2012b. Semantic expansion of tweet contents for enhanced event detection in twitter. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ASONAM ’12, pages 20–24, Washington, DC, USA. IEEE Computer Society.

Nikolaos Panagiotou, Ioannis Katakis, and Dimitrios Gunopulos. 2016. Detecting events in online social networks: Definitions, trends and challenges. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, volume 9580, pages 42–84. Springer.

Swit Phuvipadawat and Tsuyoshi Murata. 2010. Breaking news detection and tracking in twitter. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT ’10, pages 120–123, Washington, DC, USA. IEEE Computer Society.

Alberto Tonon, Philippe Cudré-Mauroux, Albert Blarer, Vincent Lenders, and Boris Motik. 2017. Armatweet: Detecting events by semantic tweet analysis. In *ESWC*.

George Valkanas and Dimitrios Gunopulos. 2013. Event detection from social media data. *IEEE Data Eng. Bull.*, 36(3):51–58.

Andreas Weiler, Michael Grossniklaus, and Marc H. Scholl. 2016. Survey and Experimental Analysis of Event Detection Techniques for Twitter. *The Computer Journal*, 60(3):329–346.

Jianshu Weng and Bu-Sung Lee. 2011. Event detection in twitter. In *ICWSM*.