Real-Time Entity-Based Event Detection for Twitter

Andrew J. McMinn(✉) and Joemon M. Jose

School of Computing Science, University of Glasgow,
Glasgow G12 8QQ, Scotland, UK
a.mcminn.1@research.gla.ac.uk, joemon.jose@glasgow.ac.uk

Abstract. In recent years there has been a surge of interest in using Twitter to detect real-world events. However, many state-of-the-art event detection approaches are either too slow for real-time application, or can detect only specific types of events effectively. We examine the role of named entities and use them to enhance event detection. Specifically, we use a clustering technique which partitions documents based upon the entities they contain, and burst detection and cluster selection techniques to extract clusters related to on-going real-world events. We evaluate our approach on a large-scale corpus of 120 million tweets covering more than 500 events, and show that it is able to detect significantly more events than current state-of-the-art approaches whilst also improving precision and retaining low computational complexity. We find that nouns and verbs play different roles in event detection and that the use of hashtags and retweets lead to a decreases in effectiveness when using our entity-base approach.

Keywords: Event detection · Social media · Reproducibility · Twitter

1 Introduction

Today, if a major event occurs, many people turn to social media services for up-to-the-second information about what is happening. Twitter is one of the most popular social media services, with over 200 million active users who make more than 500 million posts every day. Twitter makes it possible for users to post first-hand information about ongoing events in real-time, allowing Twitter to be used as a coordination tool for protests and demonstrations, with examples including the Arab Spring, and anti-government protests in Turkey.

Given this, it is not surprising that many of the most popular accounts are those which report breaking news and events. For example, the Twitter account `@breakingnews`, which aims to report breaking news in real-time, has over 6 million followers and a team of over a dozen journalists who work around the clock to monitor Twitter, but still rely on tips from over 300 other news organizations. A tool which could automatically detect, track and organize these events would be valuable to journalists and other fields, such as finance or security. However,

J. Mothe et al. (Eds.): CLEF 2015, LNCS 9283, pp. 65–77, 2015.
DOI: 10.1007/978-3-319-24027-5_6

Twitter poses a number of significant challenges which make this a hard task. The vast majority of social media content is trivial and unrelated to on-going real-world events. It is not uncommon to find a user who only posts about the food they eat or music they listen to. The low quality of social media content poses further issues; spelling and grammar errors very common, as is the use of abbreviations and acronyms. Additionally, the massive volume of data produced by social media services makes it incredibly difficult to process in real-time, and many traditional event detection approaches, such as those proposed as part of the Topic Detection and Tracking Project, fail to scale to Twitter-sized corpora. These challenges, combined with the fact that event detection and tracking is hard (even on newswire documents [3]), make it a worthy challenge.

We propose the use of named entities for the efficient and effective detection and tracking of events on Twitter. We conjecture that named entities are the building blocks of events; the people, places and organizations involved are crucial in describing an event. For example, given the event "***Hilary Mantel*** *wins the 2012* ***Man Booker Prize*** *for her novel* ***Bring Up the Bodies***", it is clear that named entities (highlighted in bold) play a crucial role in describing an event, and are often enough to decipher what happened. Our real-time approach identifies bursty named entities and uses an efficient clustering approach to detect and break events into individual topics, each of which describes a different aspect of an event. We evaluate our event detection approach on a large-scale Twitter dataset of 120 million Tweets and over 500 events, showing that our approach gives significant increases in precision and recall over current state-of-the-art approaches whilst maintaining real-time performance.

2 Background

The Topic Detection and Tracking (TDT) project aimed to produce a system that was capable of monitoring broadcast news and could produce an alert when a new event occurred. A simple nearest neighbor clustering approach was used by most TDT systems, and produced some reasonably effective systems [2–4,22]. However, TDT datasets used long newswire documents, and had several orders of magnitude fewer documents than Twitter datasets. This mean that systems were designed without regard for real-time performance or noise and spam, making them inefficient and ineffective when applied to Twitter. Despite these issues, TDT-inspired clustering models are still commonly used in event detection approaches for Twitter, although often with efficiency optimizations to cope with the increased volume of data [1,18] and additional filtering steps to remove spam and non-event clusters [6]. However, these efficiency optimizations and filtering steps often come at the cost of reduced effectiveness [1,18] or mean that significant delays must be introduced at the cost of real-time performance [6].

In recent years, interest in events (significant things that happen at some specific time and place [15]) on social media, and in particular Twitter [5], has exploded as real-time social media streams have become available for research. Hu et al. [9] demonstrated the effectiveness of Twitter as a medium for breaking

news by examining how the news of Osama bin Laden's death broke on Twitter. They found that Twitter had broken the news, and as a result, millions knew of his death before the official announcement. Kwak et al. [12] analyzed the top trending topics to show that the majority of topics (over 85%) are related to headline news or persistent news. Osborne et al. [16] measured the delay between a new event appearing on Twitter and the time taken for the same event to be updated on Wikipedia, finding that Twitter appears to be around 2 hours ahead of Wikipedia. These findings show that Twitter is a valuable resource and viable platform for the real-time detection and tracking of events and breaking news.

Although there have been many event detection approaches proposed for Twitter [1,5,6,17,18,20,21], Petrović et al. [18] were perhaps the first to propose a scalable, real-time, event detection system for Twitter. They use Locality Sensitive Hashing (LSH) to perform approximate nearest neighbor clustering in a fixed time. Recent evaluations [15] show that although the approach performs reasonably, it is susceptible to insignificant and mundane events, and has relatively low precision. Other approaches have been proposed which have high precision [6,21], however generally these approaches require significant amounts of training or curated data. Our approach achieves extremely high precision (a 100% improvement over [18]) without the need for training or curated data.

2.1 Named Entities in Events and Twitter

We believe that named entities play a key role in describing events, such as the people involved, or the location where the event took place. Without this information, or some other contextual clue, it is unreasonable to expect a person or machine to determine the specifics of an event. For example, given the document "*A bomb exploded.*", it is impossible to determine who was involved or where the event took place – we are only able to say that a bomb exploded somewhere. Only by introducing entities or other contextual information can we begin to determine the specifics of an event: "***Boko Haram*** *claims responsibility for a bomb which exploded in the northeast* ***Nigerian*** *town of* ***Potiskum.***". Given this information, we are now able to say who was involved (Boko Haram), where the event took place (Potiskum, Nigeria), and due to Twitter's real-time nature, infer with some confidence that the event took place recently.

Previous work has examined the use of named entities for event detection as part of the TDT project, with some success at improving detection performance [10,11,23]. However, these approaches were mainly adjustments to similarity measures so that named entities were given increased weight compared to other terms, and do not address the efficiency and effectiveness issues outlined in Section 2. In the context of Twitter, Choudhury and Breslin [7] examined how linguistic features and background knowledge could be used to detect specific types of sports events with high precision, however requires significant domain knowledge and large amounts of manual labeling to prepare a classifier. More similar to our work, Ritter et al. [20] used named entities, "event phrases" and temporal expressions to extract a calendar of significant events from Twitter. However, their approach requires that tweets contain temporal resolution

phrases, such as "tomorrow" or "Wednesday" to resolve between an entity and a time. This means that smaller and unexpected events, which are often the events which are of most interest, are unlikely to be detected. Our approach requires no domain knowledge or temporal phrases to perform event detection, instead relying on statistical information about common named entities, which is automatically extracted from the corpus in real-time.

Natural Language Processing (NLP) software struggles when faced with the short length and noise found in Tweets [8,13,14,20]. Several attempts have been made to address this, particularly in the tasks of Part Of Speech (POS) Tagging [8] and Named Entity Recognition (NER) [8,13,14,20]. A number of new methods have been proposed which provide significant improvements to effectiveness [13,14,20], and a number of improved models designed specifically for Twitter [8] have been released for commonly used NLP software such as the Stanford NLP Toolkit, including the GATE Twitter POS model [8].

3 Entity-Based Event Detection

In this section we describe our entity-based event detection approach. The approach comprises of 6 key stages, as shown in Figure 1. Tweets are processed in order using a pipelines architecture which allows for simple parallel processing, and with each component relying only on the output of the previous component to complete its task.

3.1 Pre-processing

Parsing and Tagging. We perform Part of Speech (POS) tagging and Named Entity Recognition (NER) on the text of each Tweet using the GATE Twitter POS model [8] which was trained using English language tweets. We extract lemmatized nouns and verbs, and named entities (persons, locations, and organizations) from each tweet.

Filtering. Event detection on Twitter relies heavily on filtering as many non-event related tweets as possible. We apply a set of filters which remove over 95% of tweets, resulting in considerably less noise, and unlike other approaches which filter after clustering [6,18], it significantly reduces the amount of data which

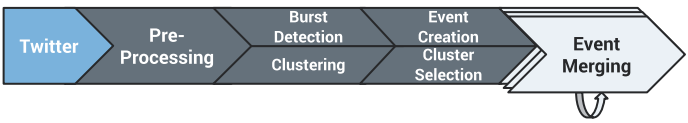

Fig. 1. The pipeline architecture and components of our approach.

needs to be processed. Our first and most aggressive filter (removing around 90% of tweets), removes Tweets which contain no named entities. As discussed in Section 2.1, we believe that named entities play a crucial role in describing events, thus do not believe that this filter significantly harms detection performance, and analyze this in Section 5.1. Furthermore, in order to efficiently cluster tweets, our clustering approach (described in Section 3.2) requires each tweet to contain at least one named entity, making this filter necessary. The second filter removes retweets, which make up approximately 30% of tweets. We examine the effect of removing retweets in Section 5.2. We also have a number of term-level filters that remove terms that are unlikely to be related to an event or that are known to be associated with spam and noise (e.g. "watch", "follow", "listen", etc.).

3.2 Clustering

Clustering is commonly used in event detection, however it is also inherently slow for large numbers of documents. We address this using the premise that tweets discussing an event must describe at least one of the named entities involved in the event, and partition tweets based upon the entities they contain, as shown in Figure 2, which reduces the computational complexity significantly. For the purpose of clustering, this can be thought of as having a unique Inverted Index for each named entity. For each named entity e in tweet d, a list of tweets D is retrieved from the inverted index for e and the maximum TF-IDF weight cosine similarity score is calculated between d and each tweet in D. If the maximum score is above a set threshold (usually in the range $0.45 - 0.55$ [18]), then d is added to the same cluster as its nearest neighbor. If the nearest neighbor does not already belong to a cluster, then a new cluster is created containing both tweets and assigned to entity e. The new tweet is then added to the inverted index for entity e. To ensure real-time performance, we limit the number of tweets that can be retrieved from an entity's inverted index to N (in our experiments, N=200 resulted in no significant differences from an unlimited number of tweets), and use only the top 10 TF-IDF weighed terms per tweet, ensuring an upper bound of $10N$ comparisons (less than 1% of tweets contain more than 10 terms).

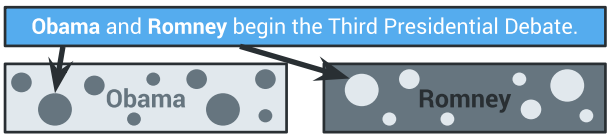

Fig. 2. Our clustering approach which partitions tweets based upon the entities they contain. The example shows how a Tweet containing both 'Obama' and 'Romney' would be put into two clusters, one for 'Obama' and one for 'Romney'.

3.3 Burst Detection

For an effective event detection approach, it is important to have a method of detecting significant events and filtering the mundane. We do this by looking for temporal bursts in the frequency of an entity, which can occur over periods ranging from a few minutes to several hours. To model this, we use a set of windows for each entity to capture their frequency over time, starting at a 5 minutes, and doubling in length up to 360 minutes (i.e. 5, 10, 20, ..., 360). We use the Three Sigma Rule as the basis for a light-weight burst detection approach, which states that a value is considered to be practically impossible if it is further than 3 standard deviations from the expected value [19]. For the windows, we maintain mean and standard deviation values, updating them periodically with the current entity frequency. It is possible to efficiently compute moving mean (μ) and standard deviation (σ) values using a set of three power sums s_0, s_1 and s_2, where s_j, μ and σ at time period $n+1$ for data series x is shown below:

$$s_{jn+1} = s_{jn} + x_{n+1}^{j} \qquad \mu = \frac{s_1}{s_0} \qquad \sigma = \sqrt{\frac{s_2 - s_1^2}{s_0}}$$

The μ and σ values for each window are updated periodically based upon the length of the window (i.e., a 5 minute window is updated every 5 minutes). When a tweet is no longer covered by the largest window it is removed from all inverted indexes. Once a tweet has been clustered and added to an entity's inverted index (as described in Section 3.2), each window is checked, and if the number of tweets in a given window is greater than $\mu + 3 \cdot \sigma$ then we say that the given window is bursty. In order to smooth and reduce noise, statistics are not updated while a window is bursting, and windows are kept in a bursting state for $1.5 \times window\_length$ after the window's statistics suggest that it has stopped bursting. This prevents large events from saturating an entity's statistics, which would make it difficult for future events to cause a burst.

3.4 Cluster Identification

Once a burst has been detected, an event is created and associated with the bursting entity for the duration of the burst (we address how an event could be associated with multiple entities in Section 3.5). However, the event does not yet have any tweets associated with it since many of the tweets posted during the burst will discuss background topics and noise. To solve this, we associate entity clusters with the event, and require that the centroid time of a cluster (i.e. the average timestamp of tweets in the cluster) is after the initial burst. This helps to ensure that clusters which discuss background topics are not included as they are likely to have existed for some time before the burst took place. A cluster's centroid time is updated as new tweets are added, ensuring that clusters which initially had a centroid time prior to the burst can still be added to an event, allowing clusters containing early reports of the event (which often occur before any burst takes places) to be included. We also require that a cluster meets a

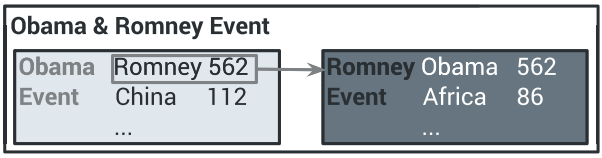

Fig. 3. An example of event merging, where two events which are happening at the same time can be merged if more than half of the tweets mention another entity.

minimum size threshold (we use 10 in our experiments) to prevent small but noisy clusters from being included. An event is kept alive as long as it has at least one bursting entity associated with it. Once all entities associated with an event have stopped bursting, the event is finalized, and no more clusters can be added to it.

3.5 Event Merging

It is common for more than a single entity to be involved in an event, such as football matches or political debates. Rather than have a single event for each entity involved in a real-world event, we attempt to automatically detect links and merge events where a link is found. If an entity is mentioned in at least 50% of tweets in an event and the mentioned entity is currently part of an another event, we merge the two events, as shown in Figure 3. This merging process can happen any number of times to produce events with many entities, and from each entity, many topics/clusters.

Entities are kept in their longest form rather than being split into individual components (e.g. 'Barack Obama', rather than 'Barack' and 'Obama'). It is unlikely that single tweet will mention both 'Barack Obama' and 'Obama', meaning that our event merging approach is unlikely to ever create a link between the two. To solve this, we perform a normalization step which splits Person names into their individual components when computing entity frequencies within events, allowing 'Barack Obama' events to be easily linked to both 'Barack' and 'Obama.'

4 Experimentation

To perform repeatable evaluations, we used the Events2012 Twitter Dataset created previously [15]. The collection provides a sample of the Twitter garden hose: 120 million tweets, covering a 28 day period, starting 10th October 2012 and ending 7th November 2012. The collection contains over 150,000 relevance judgments for over 500 events, and was created using the Wikipedia Current Events Portal[1] and 2 state-of-the-art event detection approaches, namely the

[1] http://en.wikipedia.org/wiki/Portal:Current_events

Table 1. The results from the 2 baselines (LSH & CS) and our entity-based approach when measured using Crowdsourcing (events with 30+ tweets) and automatic using the 500 events from the Events2012 collection (best run, events with 75+ tweets).

	CS	LSH	Entity (Crowd)	Entity (Auto)
Precision	53/1097 (0.048)	382/1340 (0.285)	**769/1210 (0.636)**	**181/586 (0.302)**
Recall	32/506 (0.063)	156/506 (0.308)	**194/506 (0.383)**	**159/506 (0.310)**
F1	0.054	0.296	**0.478**	**0.306**

Locality Sensitive Hashing (LSH) approach proposed by Petrović et al. [18] and the Cluster Summarization (CS) approach proposed by Aggarwal et al. [1].

Given that no event detection technique for Twitter has been robustly evaluated against a publicly available Twitter collection, the only available baselines are the LSH and CS approaches used to generate the collection. The results of the baseline approaches are taken from those given in [15]. We ran our entity based approach on the collection, treating it as a stream ordered by the creation time of each tweet. Evaluations were performed on all events with more than 30 tweets. We say that a candidate event has detected an event from the relevance judgments if at least 5% or more than 15 of the candidate's tweets match those in the relevance judgments for an event. The rationale for these choices is described in Section 5.

Although the collection contains a very large number of events and relevance judgments, we note that it does not guarantee full coverage of events, or full coverage of relevance judgments for each event. Whilst this is an issue in many IR collections, we note that the effect is more pronounced when dealing with event detection, as it is very likely that we will detect events which are not in the judgments. We verify this hypothesis and show that we detect a large number of events which are not in the relevance judgments through crowdsourcing. In order to keep the comparison between the baselines approaches and our approach as fair as possible, we used the same methodology used to gather relevance judgments for the collection. We will also make our events and judgments available as a baseline and to enhance the judgments available in the collection. We replicate the methodology described in our previous work [15], using 5 crowdsourced annotators to judge each event, and gather descriptions and category information. We perform a number of spam and quality controls, and use majority judgment for each event. Full details can be found in [15].

5 Results and Discussion

As shown in Table 1, our approach is able to significantly outperform the two baseline approaches when evaluated using Crowdsourcing and, to a lesser extent, using the test collection. Note that the crowdsourced results were only obtained for events with at least 30 tweets in order to match the evaluation carried out in [15], however the collection based evaluation was carried out on events with

Table 2. The types of event broken down by category [15] which our entity-based approach was able to detect in events with at least 30 tweets.

Category	Events	Recall
Armed Conflicts & Attacks	51	0.520
Arts, Culture & Entertainment	12	0.226
Business & Economy	9	0.391
Disasters & Accidents	13	0.448
Law, Politics & Scandals	54	0.386
Miscellaneous	4	0.190
Science & Technology	4	0.250
Sports	47	0.373

at least 75 tweets as this gave the highest F1 measure. For a direct comparison to the crowdsourced evaluation, an automatic evaluation on events with at least 30 tweets results in a precision of 0.200 and recall of 0.383, substantially below the precision of 0.636 found using crowdsourcing. We discuss possible reasons for this in Section 5.3.

Table 2 shows the breakdown of the events detected by our baseline run for events with at least 30 tweets. Our approach seems to be most effective at detecting events categorized as "Disasters & Accidents" ($R = 0.448$) and "Armed Conflicts & Attacks" ($R = 0.520$). This is extremely promising as these are the types of event that are most likely to benefit from citizen journalism and the use of social media. The ability to find and post information about these types of event can be crucial, and is one of main motivations for event detection on Twitter. Our approach also seems to be effective at detecting events categories as "Sports" ($R = 0.373$), "Business & Economy" ($R = 0.391$), and "Law, Politics & Scandals" ($R = 0.386$). Law, Politics & Scandals, as well as the Sports events make up over 50% of the total events in the collection, so given our approaches high recall, it is not surprising to find that it performs well on events in these categories. This is most likely due to a number of factors. Firstly, these types of event tend to focus on a small number of easily identified entities, such as sports teams or politicians. Secondly, these types of event are of interest to a large number of people, making them more likely to burst and be detectable, with sports events in particular being well suited to discussion on social media.

Our approach performs worst on "Miscellaneous" ($R = 0.190$), "Arts, Culture & Entertainment" ($R = 0.226$), and "Science & Technology" ($R = 0.250$) events. The low recall for science and technology events can be somewhat explained by a lack of easily detectable named entities, particularity for science events, such as "Astronomers detect what appears to be light from the first stars in the universe". Certainly, of the 21 Miscellaneous events, 10 of them have fewer than 15 tweets in the relevance judgments which contain named entities. This lack of named entities makes miscellaneous very difficult to detect for our approach, and the effect is examined in detail in Section 5.1.

5.1 Effect of Named Entities

One of the concerns using our entity-based approach is tweet recall. Since we discard any tweets without named entities we must examine the impact this has on both our tweet and event recall. Running the Stanford POS Tagger and NER over tweets from the relevance judgments from [15] shows that 47.4% of event-related tweets contain at least one named entity. This is promising, and considerably higher than the 11% of tweets that contain name entities across the collection as a whole, confirming our hypothesis that there is a relationship between entities and events. Our approach achieves a tweet recall of 0.242 across the events it detects (0.511 if we measure only against tweets which contain named entities). However, even if we were to detect every event in the collection, we could never achieve a tweet recall above 47.6%. Some of this is likely down to the difficulty of NER on Twitter, as noted by Li et al. [13], and could be improved with better NER models for Twitter. However, it is likely that the majority of tweets do not contain named entities, meaning that we must consider the effect this has on detection effectiveness – if an event has very few or no tweets with named entities then our approach will be unable to detect them.

Of the 506 events[15], 14 have fewer than 5 relevance judgments, 42 have fewer than 15, and 72 have fewer than 30. In addition, 41 events in the relevance judgments have fewer than 5 tweets with entities, 109 events have fewer than 15, and 163 have fewer than 30. For those 41 events with fewer than 5 tweets containing entities, even if our system was to perform perfectly, we would be unable to detect them – that accounts for just over 8% of all the events in the collection. However, given than these events on average contain just 32 tweets, it seems unlikely that they are of any real-world significance.

5.2 Nouns, Verbs, Hashtags and Retweets

Table 3 shows the effect of using different terms combinations for clustering. Note that for verb only clustering, named entities were still used, despite being classified as proper nouns. This is because our clustering and event merging approaches require them to work, however we feel that conclusions drawn from the result of this run are still valid and can be used to provide insight. The use of only nouns gives the highest recall but the lowest precision ($F1 = 0.249$), whereas using verbs only results in the lowest recall but the highest precision.

Table 3. The effect of using different combinations of nouns (NN), verbs (VB) and hashtags (HT) as terms for clustering on events with at least 30 tweets.

POS	Precision	Recall	F1
NN Only	242/1324 (0.183)	198/506 (0.391)	0.249
VB Only	196/912 (0.215)	165/506 (0.326)	0.259
NN, VB	242/1210 (0.200)	194/506 (0.383)	0.263
NN, VB, HT	232/1174 (0.198)	192/506 (0.379)	0.260

Using both nouns and verbs seems to take best the characteristic of both, giving the highest F1 measure. The high recall associated with nouns fits with our hypothesis that events are about entities, as named entities are proper nouns, and entity classes (i.e. city, person, plant) are common nouns. If nouns had not been used to describe these events then we would not have been able to detect them. This is again reflected in the low recall when using only verbs, and we suspect that had named entities (i.e. proper nouns) not been used, then the recall would be even lower.

The use of Hashtags seems to cause a small but insignificant reduction in both precision and recall, a somewhat unexpected result, as Hashtags are commonly thought to be very good indicators for the topic of a tweet. We hypothesis that this is due to the specificity of named entity, and by requiring every tweet to contain a named entity, we are removing the topical uncertainty and rendering Hashtags redundant as an indicator of topic. The use of retweets has a significant impact, reducing precision from 0.200 to 0.063. The use of retweets does provide a small, but insignificant increase in recall (0.390), and can likely be attributed to a 60% increase in the average number of tweets per event from 125 to 198, creating many events with more than 30 tweets. These findings are somewhat unsurprising as retweets have previously been associated with the spread of spam and require little effort to produce.

5.3 Evaluation Measures

Event detection on Twitter is a relatively new task, with very little work looking at how to perform reproducible and reliable evaluations. The work in this paper presents, to the best of our knowledge, the first evaluation of an event detection approach on a large-scale, publicly available dataset, and as such it is important to examine our choice of evaluation measures and thresholds. In this work, we required that at least 5% or at least 15 tweets in a candidate event must be relevant to a single event from the relevance judgments for it to be considered detected. The rational for this is two-fold. Firstly, it is impossible that the collection has judgments for every event which occurred over the 28 days it covers. Even for events which are in the collection, it is unlikely to have complete coverage of all relevant tweets. Secondly, because automated methods were used to generate the events, each with differing levels of granularity, there are a number of events in the judgments where only part of an event has been detected (for example, a single goal in a football matcher, rather than the football match itself). This means that a high threshold will make it difficult for an event to be relevant if the system has detected a "full" event rather than the specific sub-event which the collection has judgments for. While this may seem like a somewhat low threshold, we feel that it is reasonable, and by comparing the precision of our approach using the collection (0.200) and using crowdsourcing (0.636), it is clear that it does not result in an overestimate of precision.

6 Conclusion

In this paper, we proposed a novel, efficient, real-time, event detection approach for Twitter using the role that named entities play in events. We used a clustering technique which partitions tweets based upon the entities they contain, burst detection and cluster selection techniques to extract clusters related to ongoing real-world events. We demonstrated that our approach is able to outperform state-of-the-art approaches, whilst retaining a very low computational complexity and guaranteeing real-time performance. We found that nouns and verbs have significant roles in determining recall and precision respectively, and that Twitter-specific features seem to have either no effect or a detrimental effect on detection performance when using our entity-based approach.

References

1. Aggarwal, C.C., Subbian, K.: Event detection in social streams. In: Proc. of SDM Conference (2012)
2. Allan, J., Harding, S., Fisher, D., Bolivar, A., Guzman-Lara, S., Amstutz, P.: Taking topic detection from evaluation to practice. In: HICSS 2005. IEEECS, Washington, D.C. (2005)
3. Allan, J., Lavrenko, V., Jin, H.: First story detection in TDT is hard. In: CIKM 2000, pp. 374–381. ACM, New York (2000)
4. Allan, J., Lavrenko, V., Malin, D., Swan, R.: Detections, bounds, and timelines: UMass and TDT-3. In: TDT-3 Workshop (2000)
5. Atefeh, F., Khreich, W.: A survey of techniques for event detection in twitter. Computational Intelligence (2013)
6. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: real-world event identification on twitter. In: ICWSM 2011 (2011)
7. Choudhury, S., Breslin, J.G.: Extracting semantic entities and events from sports tweets. In: Proceedings of #MSM2011 at ESWC (2011)
8. Derczynski, L., Ritter, A., Clark, S., Bontcheva, K.: Twitter part-of-speech tagging for all: overcoming sparse and noisy data. In: ICRA-NLP (2013)
9. Hu, M., Liu, S., Wei, F., Wu, Y., Stasko, J., Ma, K.-L.: Breaking news on twitter. In: CHI 2012. ACM, New York (2012)
10. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: SIGIR 2004, pp. 297–304. ACM, New York (2004)
11. Kumaran, G., Allan, J.: Using names and topics for new event detection. In: HLT 2005, pp. 121–128. ACL, Stroudsburg (2005)
12. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: WWW 2010. ACM, New York (2010)
13. Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., Lee, B.-S.: Twiner: named entity recognition in targeted twitter stream. In: SIGIR (2012)
14. Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets. In: HLT 2011. ACL, Stroudsburg (2011)
15. McMinn, A.J., Moshfeghi, Y., Jose, J.M.: Building a large-scale corpus for evaluating event detection on twitter. In: CIKM 2013. ACM (2013)
16. Osborne, M., Petrovic, S., McCreadie, R., Macdonald, C., Ounis, I.: Bieber no more: first story detection using twitter and wikipedia. In: SIGIR 2012 Workshop TAIA (2012)

17. Ozdikis, O., Senkul, P., Oguztzn, H.: Semantic expansion of tweet contents for enhanced event detection in twitter. In: ASONAM. IEEE CS (2012)
18. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In HLT 2010. ACL (2010)
19. Pukelsheim, F.: The three sigma rule. The American Statistician **48**(2), 88–91 (1994)
20. Ritter, A., Mausam, Etzioni, O., Clark, S.: Open domain event extraction from twitter. In: Proceedings of ACM SIGKDD 2012. ACM (2012)
21. Sankaranarayanan, J., Samet, H., Teitler, B., Lieberman, M., Sperling, J.: Twitterstand: news in tweets. In: ACM SIGSPATIAL 2009 (2009)
22. Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: SIGIR 1998, pp. 28–36. ACM, New York (1998)
23. Yang, Y., Zhang, J., Carbonell, J., Jin, C.: Topic-conditioned novelty detection. In: ACM CIKM 2002, pp. 688–693 (2002)