

MaTED: Metadata-assisted Twitter Event Detection System

Abhinay Pandya¹[0000-0002-3261-2920], Mourad Oussalah², Panos Kostakos¹[0000-0002-8545-599X], and Ummul Fatima¹

¹ Center for Ubiquitous Computing, Faculty of ITEE, University of Oulu, Finland.
<http://ubicomp.oulu.fi>

² Center for Machine Vision and Signal Analysis, Faculty of ITEE, University of Oulu, Finland.

<http://www.oulu.fi/cmvs>

{[abhinay.pandya](mailto:abhinay.pandya@oulu.fi), [mourad.oussalah](mailto:mourad.oussalah@oulu.fi), [panos.kostakos](mailto:panos.kostakos@oulu.fi), [ummul.fatima](mailto:ummul.fatima@oulu.fi)}@oulu.fi

Abstract. Due to its asynchronous message-sharing and real-time capabilities, Twitter offers a valuable opportunity to detect events in a timely manner. Existing approaches for event detection have mainly focused on building a temporal profile of named entities and detecting unusually large bursts in their usage to signify an event. We extend this line of research by incorporating external knowledge bases such as DBPedia, WordNet; and exploiting specific features of Twitter for efficient event detection. We show that our system utilizing temporal, social, and Twitter-specific features yields improvement in the precision, recall, and DERate on the benchmarked Events2012 corpus compared to the state-of-the-art approaches.

Keywords: Twitter · Event Detection · Dbpedia · Microblogging · Social Media

1 Introduction

Social media serve as important *social sensor* to capture the zeitgeist of the society. With real-time, online, asynchronous message-sharing supporting text, audio, video and images, these platforms offer valuable opportunities to detect events such as natural disasters and terrorist activity in a timely manner. Twitter is certainly a leader among microblogging platforms. With over 1.3 billion users and about 500 million messages (called tweets) posted per day and over 15 billion API calls per day, Twitter provides a massive source to detect event occurrences as they happen. Tweets are restricted to be maximum 280 characters in length and it is this design choice that enables information sharing extremely fast and in real-time. Owing to this scale and speed of information updates, Twitter is at the focus of attention to be monitored for new event detection. Also, Twitter's social network features allow user interactions which helps further in gaining an insight on an event's reception by people (by analysing their sentiments/opinions expressed in their tweets).

While Twitter analytics in general may entail a comprehensive multi-modal approach, for example, to also harness relevant information from media (images, videos, audio), we argue that since the textual part of the message is authored by the creator of the message, it is more reliable, authentic, and credible personal source to gain insights on the information shared. However, while analyzing language is as such challenging because of the inherent lack of structure in its expression, Twitter exacerbate this by supporting short message text, slang and emojis, user-created meta-information tags (hashtags), and URLs. Traditional methods in NLP were designed to work with large discourses and perform poorly on social media text. This is owing to the following:

- Tweets are short mainly because of the restrictions imposed by the underlying platform (i.e. Twitter), but sometimes also under the influence of the cultural norms of the cyberspace ³;
- Tweets are often ungrammatical. Under the restriction of typing short messages from a mobile phone, grammar often takes a back seat. Also, because of the growth of social media encompassing the global scale, significant number of users are non-native speakers of English;
- Tweets are highly contextual, i.e., the message cannot be understood without the context – geopolitical, cultural, topical (current affairs), and even conversational context (e.g. replies to previous messages);
- Tweets use a lot of slang and emojis; exacerbating this, new slang words and emojis are invented and popularized every day and their meanings are volatile.

Indeed, tweets contain polluted content [18], and rumors [6], which negatively affect the performance of the event detection algorithms. Besides, only very few tweets actually carry a message about a newsworthy event [13]. These limitations motivate the current work, which aims to develop an enhanced Twitter event detection system that integrates external sources of information such as DBPedia [4] and WordNet [7]. Topic Detection and Tracking (TDT) project [2] defines an event as “some unique thing that happens at some point in time”. [5] 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 use these definitions as a guideline in developing our system of event detection from Twitter. We exploit specific features of Twitter that allow users to share others’ tweets by re-tweeting them, quoting them, liking them; allows users to embed hashtags, URLs.

Much like many event detection methods([19],[10],[31]), our system is based on identifying important phrases from individual tweets and creating a temporal profile of these phrases to identify if they are bursty enough to signify an occurrence of an event. However, our system differs from the state-of-the-art in the following:

- We utilize DBPedia to efficiently identify named entities of import.

³ https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html

- We use WordNet to assist in identifying event-specific words and phrases.
- We expand the URLs embedded in tweets to find important phrases from the titles of the web pages that these URLs point to.
- We harness the meta-data from Twitter such as ‘quoted status’, ‘retweet’, ‘liked’, ‘in reply to’, and also, the social network of the Twitter user.

The rest of the paper is organized as follows. In Section 2, we discuss the work closely related to ours. Section 3 describes our system for Twitter event detection. Section 4 explains our experimental setup and results followed by discussion. Finally, in Section 5, we present our conclusions.

2 Related Work

Event detection from Twitter has been extensively studied in the past as evidenced by the rich body of work. [14], [3], [25], [30], among others, presented surveys on techniques for event detection from Twitter classifying them according to the event type (specified or unspecified events), detection method (supervised or unsupervised learning), and detection task (new event detection or retrospective event detection). However, most of the techniques described in these surveys suffer from rigorous evaluation. The major obstacle in measuring the accuracy and performance of an event detection methods is the lack of large-scale, benchmarked corpora. Some authors have created their own manually annotated datasets and made them available publicly [23], [26].

Our work focuses on unsupervised, unspecified event detection retrospectively from a large body of tweets. Among many approaches to event detection from Twitter such as keyword volume approach, topic modeling, and sentiment analysis based methods, our work is based on keyphrase/segment detection and tracking which aim to identify keyphrases/segments whose occurrences grow unusually within the corpus [17], [16], [8], [11], [12], [29], [9]. Some of the most related works to ours are by [31], [19], [1], [27].

EDCoW[31] proposes a three-step approach. First, a wavelet transform and auto-correlation are applied to measure the bursty energy of each word and words associated with high energies are retained as event features. Then they measure the similarity between each pair of event features by using cross correlation. At last, modularity-based graph partitioning is used to detect the events, each of which contains a set of words with high cross correlation.

[19] presented a system called Twevent that analyzes the tweets by breaking them into non-overlapping segments and subsequently identifying bursty segments. These bursty segments are then clustered to obtain event-related segments.

[23] contributed a large manually labeled corpus of 120 million tweets containing 506 events in 8 categories. They used Locality Sensitive Hashing (LSH) technique followed by cluster summarization and employed Wikipedia as an external knowledge source.

[1] employed statistical analysis of historical usage of words for finding bursty words – those with burstiness degree above two standard deviation from the

mean are selected clustered. However, their method was used to find localized events only.

[10] propose mention-anomaly based approach incorporating social aspect of tweets by leveraging the creation frequency of mentions that users insert in tweets to engage discussion. [22] advocated the importance of named entities in Twitter event detection. They 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.

Recently, [28] employed extracting a structured representation from the tweets' text using NLP, which then was integrated with DBpedia and WordNet in an RDF knowledge graph. Their system enabled security analysts to describe the events of interest precisely and declaratively using SPARQL queries over the graph.

3 Our System

Our system, Metadata-assisted Twitter Event Detection (MaTED), is an extension of a previous work Twevent [19]; however, our system makes use of several other features of Twitter ignored in previous research.

Figure 1 shows the architecture of MaTED which consists of four components: i) detection of important phrases from tweets; ii) creating temporal profiles of these phrases to identify bursty phrases; iii) clustering bursty phrases with an aim to group related phrases about an event, and iv) characterizing an event from the clusters obtained above. We parse the tweet JSON object after receiving it from a stream, and the first component of our system identifies important segments/phrases not just from the tweet text but also from the titles of the webpages that URL links in the tweet points to. Since event-related phrases are mostly named entities, we harness the DBpedia to extract such phrases. The resultant phrases along with tweet timestamps are then fed to the next component of our system which estimates their burstiness behavior using statistical modeling of their occurrence frequency. Subsequently, we group the event-related phrases using a graph-based clustering algorithm. In the rest of this section, we describe each component in detail following the order of their usage in our framework.

3.1 Identifying important phrases from tweets

In this component, we parse the tweet JSON object to obtain tweet text, hashtags, URLs, user mentions and other available metadata. We then create a set of phrases / keywords to be monitored consisting of the following items:

- List of named entities obtained after inputting the tweet text (after pre-processing and cleaning) to DBpedia Spotlight [24] web service.
- List of wordings related to action or activity present in the original tweet message. These are identified as words that are either a direct or indirect hyponym of 'event.n.01' synset of WordNet.

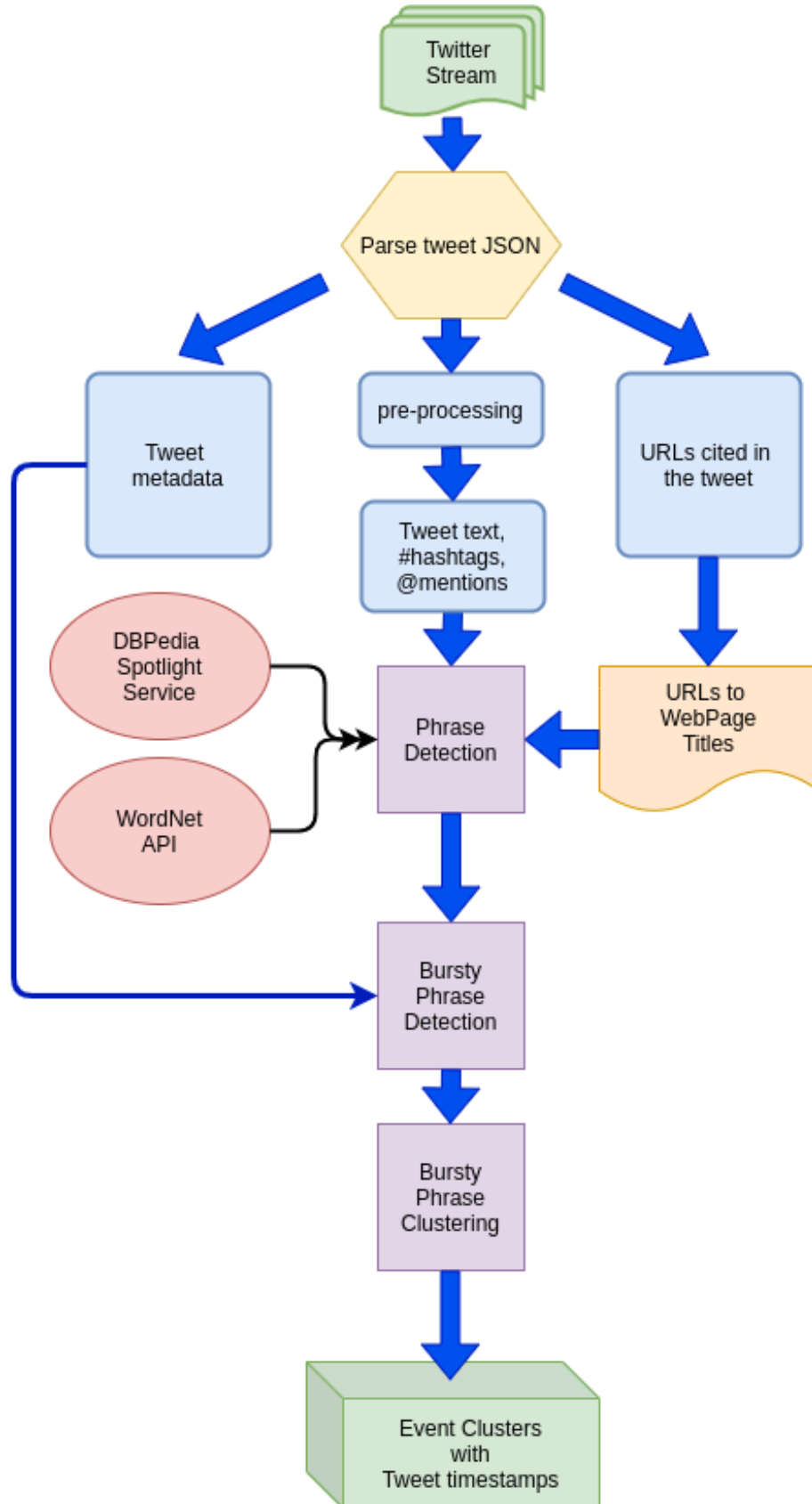


Fig. 1. MaTED system architecture

- List of hashtags included in the tweet.
- For each URL that is cited in the tweet, we obtain the title text of the webpage that URL is pointing to. We submit this title text to our locally running DBPedia Spotlight web service to obtain named entities and include these into the list of items to be monitored.
- For each user mention, we include the 'name' the user mention handle is associated to.

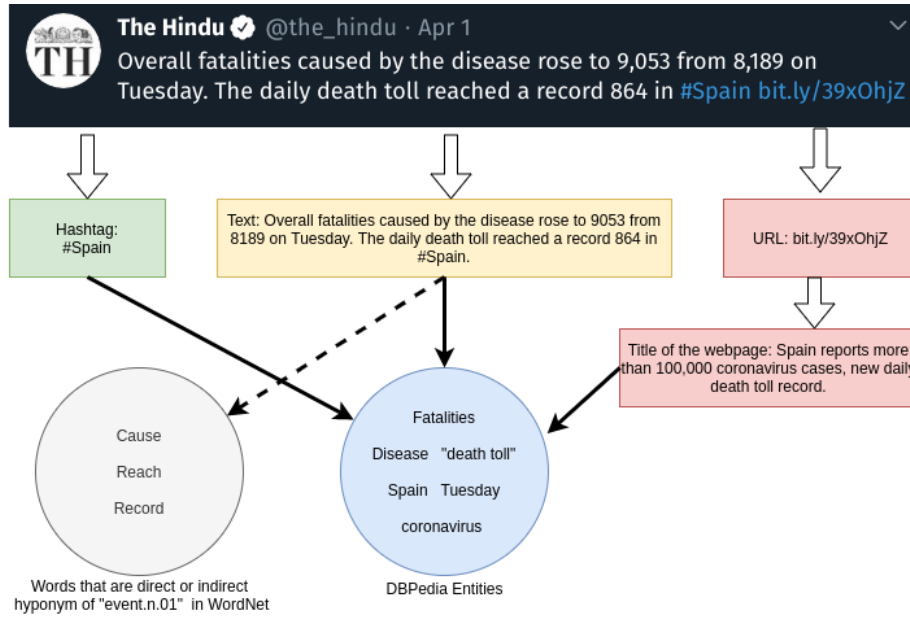


Fig. 2. An example of phrase extraction from tweets.

Figure 2 illustrates the process. The tweet text “*Overall fatalities caused by the disease rose to 9053 from 8189 on Tuesday. The daily death toll reached a record 864 in Spain.*” does not mention *coronavirus* which is important entity for event detection, tracking and monitoring. Our system fetches and processes the title of the webpage linked to the URL cited in the tweet (bit.ly/39xOhjZ). Concurrently, finding words from the tweet text that are direct or indirect hyponyms of the event.n.01 synset of the WordNet finds important words to track/monitor (‘cause’, ‘reach’, and ‘record’).

3.2 Extracting bursty phrases

After creating a set of phrases from the dataset as indicated above, using a model proposed by Tveit [19], we find bursty phrases potentially indicative of an

event. However, our model includes several other factors in finding burstiness score of a phrase than considered in [19]. Below we outline our method.

Let N_t denote the number of tweets published within the current time window t and $n_{i,t}$ be the number of tweets containing phrase i in t . The probability of observing i with a frequency $n_{i,t}$ can be modeled by a Binomial distribution $B(N_t, p_i)$ where p_i is the expected probability of observing phrase i in a random time window. Since N_t is very large in case of Twitter stream, this probability distribution can be approximated by a Normal distribution p with parameters $E[i|t] = N_t \times p_i$ and $\sigma(i|t) = \sqrt{N_t \times p_i \times (1 - p_i)}$.

We consider a phrase i as bursty if $n_{i,t} \geq E[i|t]$. Using the formula for the burstiness probability $P_b(i, t)$ for phrase i in time window t defined by [19]:

$$P_b(i, t) = S\left(10 \times \frac{n_{i,t} - (E[i|t] + \sigma[i|t])}{\sigma[i|t]}\right) \quad (1)$$

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

In addition to finding the importance of a phrase based on how many times it was used in the given time window, we further assign various weight values based on *who* authored the phrase, *how many times it was retweeted, quoted, liked, replied to*. More formally, let $u_{i,t}$ denote the number of distinct users authoring phrase i in time window t . Let retweet count of a phrase i in t be $rt_{i,t}$ which corresponds to the sum of the retweet counts of all tweets containing i in t . Similarly, let $l_{i,t}$ be the liked count, $q_{i,t}$ be the quoted count, and $rp_{i,t}$ be the ‘replied to’ count. Also, in order to assign an importance degree to the phrases used by those Twitter users who have a significant following, we assign the weight $fc_{i,t}$ as the follower count which is the sum of the follower count of all users using phrase i in t . Incorporating all the above, the burstiness weight $w_b(i, t)$ for a phrase i in t can be defined as:

$$w_b(i, t) = P_b(i, t) \cdot \log(u_{i,t}) \cdot \log(rt_{i,t}) \cdot \log(l_{i,t}) \cdot \log(q_{i,t}) \cdot \log(rp_{i,t}) \cdot \log(\log(fc_{i,t})) \quad (2)$$

After finding the burstiness weight for all phrases, the top K are selected in decreasing order of weights. Empirically, we find that decreasing K results in low recall, while increasing K brings in a significant noise. [19] suggest an optimal value of K is set to $\sqrt{N_t}$.

3.3 Clustering bursty phrases

We adopt the approach by [19] without any modification to group bursty phrases to derive event-related clusters. Each time window is evenly split into M sub-windows $t = \langle t_1, t_2, \dots, t_M \rangle$. Let $n_t(i, m)$ be the tweet frequency of phrase i in the subwindow t_m and $T_t(i, m)$ be the concatenation of all the tweets in the subwindow t_m that contain phrase i . The similarity $sim_t(i_a, i_b)$ between phrases i_a and i_b in time window t is calculated as follows:

$$sim_t(i_a, i_b) = \sum w_t(i_a, m)w_t(i_b, m) \times sim(T_t(i_a, m), T_t(i_b, m)) \quad (3)$$

where $\text{sim}(T_t(i_a, m), T_t(i_b, m))$ is TF-IDF similarity between tweets $T_t(i_a, m)$ and $T_t(i_b, m)$; and $w_t(i_a, m)$ is the fraction of frequency of segment i_a in the time subwindow t_m as calculated as follows:

$$w_t(i, m) = \frac{n_t(i, m)}{n_{(i,t)}} \quad (4)$$

Using the above similarity measures, all the bursty phrases are clustered using a graph-based clustering algorithm [15]. In this method, all bursty phrases are considered as nodes and initially, all nodes are disconnected. An edge is added between phrases i_a and i_b if k -Nearest neighbors of i_a contain i_b and vice versa. All connected components of the resultant graph are considered as candidate event clusters. Each connected component is essentially a set of phrases which are related to a single event. Disconnected nodes (phrases) are discarded as non-significant.

3.4 Event Characterization

We characterize an event as a group of phrases associated to it. To visualize this, we adopt the approach by MABED[10]. An interface is designed to allow us to visualize the list of relevant tweets defining the event by ‘clicking’ on the event name.

4 Experiments and Results

We use the corpus collected by [23] Events2012 which contains 120 million tweets and 506 labeled events. These tweets were collected from Oct 10 - Nov 7, 2012 and were filtered to remove tweets containing more than 3 hashtags, 3 user mentions, or 2 URLs discarding them as spam [6]. However, not all tweets were available due to some users’ data being not available because of account inactivation, privacy mode setting changes, etc. Our final dataset contain ~ 38 million tweets of which 127,356 are related to events. It should be noted that these tweets are limited to maximum 140 characters in length since the increased length (up to 280 characters) was introduced in late 2017. In Table 1 we show some of the important events in the Events2012 [23] dataset which have more than 1,500 tweets associated to them.

4.1 Pre-processing

We perform the following steps sequentially as preprocessing the tweet text and the titles of the webpages linked by the cited URLs in the tweet:

1. We use the Stanford tokenizer⁴ to tokenize the tweets.

⁴ <https://nlp.stanford.edu/software/tokenizer.htm>

Table 1. Top events in Events2012 dataset. Each with more than 1500 tweets associated to it.

Event ID	Manually-annotated event description	#tweets
8	During US presidential debate, President Barack Obama tells candidate Mitt Romney he is the last person to get tough on China.	12805
1	12 Oct 2012 – Paul Ryan spoke for 40 of the 90 minutes during Thursday night’s vice presidential debate and managed to tell at least 24 myths during that time	6894
11	Barack Obama and Mitt Romney went head-to-head in the final Presidential Debate. Romney said no government that makes businesses successful!	5243
22	The Redbull Stratos jump was a space diving event. Felix Baumgartner flew many miles into the air above the southwestern U.S. and then jumped, breaking several world records.	2967
52	People react to incoming election results, threatening to leave the country if their favored candidate does not win.	2289
14	In Major League Baseball, the San Francisco Giants defeat the Detroit Tigers in game four.	1796

- Use of words like *coooooooll*, *awesommmme*, are sometimes used in tweets to emphasize emotion. We use a simple trick to normalize such occurrences. Let n denote the number of such letters that have three or more consecutive occurrences in a given word. We first replace three or more consecutive occurrences of the same character with two occurrences. Then we generate $\binom{n}{2}$ prototypes that are at edit distance 1 (only delete operation, deleting only repeated character) and look for this prototype in the dictionary to find the word. For example, *cooooooolllll* \rightarrow *cooll* \rightarrow *cool*.
- We use an acronym dictionary from an online resource⁵ to find expansions of the tokens such as *gr8*, *lol*, *rotfl*, etc.

After pre-processing, to obtain a list of named entities, we submit the text to the DBPedia Spotlight [24] web service. We chose DBPedia over other named-entity recognition software (such as Stanford NER [21], OpenNLP [32], NLTK [20], etc.) since employing such tools yields phrases that induce noise in the resulting system.

4.2 Baseline Methods

We compare our approach with closely related works: EDCoW [31], Twevent [19], NEED [22], and MABED [10]. Table 2 shows the comparative performance of our system to selected state-of-art approaches. For MABED, we modified their online available code to include hashtags, instead of user mentions to measure anomaly (ref as MABED+ht). Also, for our system, in order to observe the effect of WordNet words related to events, we conducted two sets of experiments where

⁵ <http://www.noslang.com>

MaTED-WN is the system without using WordNet words. We share our source code and dataset used online.

4.3 Evaluation Metrics

As evaluation metrics, we used precision, recall, and DERate (duplicate event rate, proposed by [19]). *Precision* conveys the fraction of the detected events that are related to a realistic event. *Recall* indicates the fraction of events detected from the manually labeled ground truth set of events. However, if two detected events are related to the same realistic event within the same time window, then both are considered correct in terms of precision, but only one realistic event is considered in counting recall. Therefore, [19] defined a metric DERate to denote the fraction of events (in percentage values) are dublicately detected among all events detected.

4.4 Results and Discussion

Table 2 shows the results we obtained compared to the baseline method.

Table 2. Results on Events2012 dataset

Approach	EDCoW	Twevent	NEED	MABED	MABED+ht	MaTED-WN	MaTED
Precision	0.42	0.81	0.61	0.68	0.74	0.83	0.79
Recall	0.38	0.73	0.41	0.43	0.55	0.79	0.82
DERate(%)	-	15.6	19.4	17.4	16.7	14.7	14.3

Table 3 shows some of the events detected by MaTED that were not detected by any of the above systems.

Table 3. A sample of events that only our system could extract.

Date	Event	Phrases
17.10.2012	Alpha Centauri BB planet discovered	[solar system],[centauri],[earth],[exoplanet]
16.10.2012	Ford recalls Fiesta because of airbag issue	[Ford],[Fiesta],[airbag],[fault]
15.10.2012	Amanda Todd suicide because of cyber-bullying	[Amanda Todd],[bullying],[suicide],[RIP]

Several parameters impact the performance of the resulting system and the results shown in Table 2 are obtained by an optimal combination of them. It is evident from Table 2 that the performance of existing bursty segment detection based systems is enhanced by including social and Twitter-specific features incorporated in our system. Especially, we notice a significant improvement in

recall by including title texts of the web pages pointed to by the URLs in the tweets. A tweet is often a comment on the web page that is shared and therefore, by including the title text, the system incorporates a better context for the tweet. Further, because of misspellings, DBpedia Spotlight sometimes fail to find the named entity and in such cases, tracking event specific words from WordNet (total 7878 words) helps identify an event. For example, in Table 3, the event on 16.10.2012 about Ford would be missed if the word ‘fault’ was not included in the list of important key-phrases to be considered. Better results are observed for MABED+ht as opposed to the original MABED owing to the fact that hashtags are better indicators of events than user mentions. We also noticed that many events were not reported in the crowd-sourced ground-truth Events 2012 corpus. Event on 15.10.2012 about Amanda Todd suicide is one example of many events we found which were not included in the corpus.

5 Conclusion

In this paper, we build on an existing system Twevent [19] by incorporating external knowledge bases of DBpedia and WordNet together with exploiting user’s mentions and hashtags contained in Twitter messages for efficient event detection. We examined the effect of adding novel features incrementally and concluded that our model outperforms the state-of-the-art on the benchmarked Events2012 corpus. Future research includes investigating usage of distributed semantics (e.g., word embeddings) incorporated in a larger framework of a deep learning inspired model towards achieving higher accuracy on event detection from a massive-scale collection of social media messages.

6 Acknowledgment

This work is partly supported by EU YoungRes project (#823701) on polarization detection.

References

1. Abdelhaq, H., Sengstock, C., Gertz, M.: Eventweet: Online localized event detection from twitter. *Proceedings of the VLDB Endowment* **6**(12), 1326–1329 (2013)
2. Allan, J.: *Topic detection and tracking: event-based information organization*, vol. 12. Springer Science & Business Media (2012)
3. Atefeh, F., Khreich, W.: A survey of techniques for event detection in twitter. *Computational Intelligence* **31**(1), 132–164 (2015)
4. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: *Dbpedia: A nucleus for a web of open data*. In: *The semantic web*, pp. 722–735. Springer (2007)
5. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: Real-world event identification on twitter. In: *Fifth international AAAI conference on weblogs and social media* (2011)

6. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. pp. 675–684. ACM (2011)
7. Fellbaum, C.: Wordnet. The encyclopedia of applied linguistics (2012)
8. Fung, G.P.C., Yu, J.X., Yu, P.S., Lu, H.: Parameter free bursty events detection in text streams. In: Proceedings of the 31st international conference on Very large data bases. pp. 181–192. VLDB Endowment (2005)
9. Goorha, S., Ungar, L.: Discovery of significant emerging trends. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 57–64. ACM (2010)
10. Guille, A., Favre, C.: Mention-anomaly-based event detection and tracking in twitter. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014). pp. 375–382. IEEE (2014)
11. He, Q., Chang, K., Lim, E.P.: Analyzing feature trajectories for event detection. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 207–214. ACM (2007)
12. He, Q., Chang, K., Lim, E.P., Zhang, J.: Bursty feature representation for clustering text streams. In: Proceedings of the 2007 SIAM International Conference on Data Mining. pp. 491–496. SIAM (2007)
13. Hurlock, J., Wilson, M.L.: Searching twitter: Separating the tweet from the chaff. In: Fifth International AAAI Conference on Weblogs and Social Media (2011)
14. Imran, M., Castillo, C., Diaz, F., Vieweg, S.: Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)* **47**(4), 1–38 (2015)
15. Jarvis, R.A., Patrick, E.A.: Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on computers* **100**(11), 1025–1034 (1973)
16. Kleinberg, J.: Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery* **7**(4), 373–397 (2003)
17. Kontostathis, A., Galitsky, L.M., Pottenger, W.M., Roy, S., Phelps, D.J.: A survey of emerging trend detection in textual data mining. In: Survey of text mining, pp. 185–224. Springer (2004)
18. Lee, K., Eoff, B.D., Caverlee, J.: Seven months with the devils: A long-term study of content polluters on twitter. In: Fifth International AAAI Conference on Weblogs and Social Media (2011)
19. Li, C., Sun, A., Datta, A.: Twevent: segment-based event detection from tweets. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 155–164. ACM (2012)
20. Loper, E., Bird, S.: Nltk: the natural language toolkit. arXiv preprint cs/0205028 (2002)
21. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp. 55–60 (2014)
22. McMinn, A.J., Jose, J.M.: Real-time entity-based event detection for twitter. In: International conference of the cross-language evaluation forum for european languages. pp. 65–77. Springer (2015)
23. McMinn, A.J., Moshfeghi, Y., Jose, J.M.: Building a large-scale corpus for evaluating event detection on twitter. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 409–418. ACM (2013)
24. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th international conference on semantic systems. pp. 1–8. ACM (2011)

25. Panagiotou, N., Katakis, I., Gunopulos, D.: Detecting events in online social networks: Definitions, trends and challenges. In: *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pp. 42–84. Springer (2016)
26. Petrovic, S.: Real-time event detection in massive streams (2013)
27. Phuvipadawat, S., Murata, T.: Breaking news detection and tracking in twitter. In: *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. vol. 3, pp. 120–123. IEEE (2010)
28. Tonon, A., Cudré-Mauroux, P., Blarer, A., Lenders, V., Motik, B.: Armatweet: detecting events by semantic tweet analysis. In: *European Semantic Web Conference*. pp. 138–153. Springer (2017)
29. Wang, X., Zhai, C., Hu, X., Sproat, R.: Mining correlated bursty topic patterns from coordinated text streams. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 784–793. ACM (2007)
30. Weiler, A., Grossniklaus, M., Scholl, M.H.: Survey and experimental analysis of event detection techniques for twitter. *The Computer Journal* **60**(3), 329–346 (2017)
31. Weng, J., Lee, B.S.: Event detection in twitter. In: *Fifth international AAAI conference on weblogs and social media* (2011)
32. Wilcock, G.: Text annotation with opennlp and uima (2009)