Sentiment-Based Event Detection in Twitter

Georgios Paltoglou
Faculty of Science and Engineering, University of Wolverhampton, Wulfruna Street, Wolverhampton WV1 1LY, United Kingdom. E-mail: g.paltoglou@wlv.ac.uk

The main focus of this article is to examine whether sentiment analysis can be successfully used for "event detection," that is, detecting significant events that occur in the world. Most solutions to this problem are typically based on increases or spikes in frequency of terms in social media. In our case, we explore whether sudden changes in the positivity or negativity that keywords are typically associated with can be exploited for this purpose. A data set that contains several million Twitter messages over a 1-month time span is presented and experimental results demonstrate that sentiment analysis can be successfully utilized for this purpose. Further experiments study the sensitivity of both frequency- or sentiment-based solutions to a number of parameters. Concretely, we show that the number of tweets that are used for event detection is an important factor, while the number of days used to extract token frequency or sentiment averages is not. Lastly, we present results focusing on detecting local events and conclude that all approaches are dependant on the level of coverage that such events receive in social media.

Introduction

Twitter has become one of the most popular social media networks in recent years. Its micro-blogging format has allowed the website to become a reference point for world-wide events and discussions, leading even established news agencies to create and promote their own Twitter accounts[1]. In this article, we define the term *event* as a "noteworthy happening" (Merriam-Webster dictionary), which may have taken place either offline (e.g., the Arab Spring[2]) or online (e.g., world-wide web access coming to a halt after the death of Michael Jackson[3]).

The ability to detect such topics[4] in a timely manner is of significant importance for a number of reasons; in reputation monitoring applications, such a capability can give interested parties, such as companies, organizations, etc., time to respond effectively before emerging issues escalate. In humanitarian crisis situations, such as earthquakes or floods, this capability can provide organizations more flexibility to address the problems in a more organized and focused manner. In politics, the capability to quickly recognize emerging issues can significantly help parties address them before they affect their voter base.

In this article, we hypothesize that apart from increases in the frequency of occurrence of keywords, which have been shown in the past to be good predictors of significant events (Aiello et al., 2013; Kleinberg, 2003; Thelwall, Buckley, & Paltoglou, 2011), significant changes in the overall sentiment that keywords are typically associated with can also be effectively used to detect such events. Our approach can have significant advantages over frequency-based solutions, especially in real-world applications. That is because, in most operational settings, interested parties have access only to a sample of tweets, via the service's "Streaming application programming interface (API)," rather than the complete Twitter data stream, named Firehose[5]. Nonetheless, recent work by Morstatter, Pfeffer, Liu, and Carley (2013), that compares the data obtained from those two streams, reports that when there is an increase in the number of tweets that match some criteria in the Firehose stream, the Streaming API's coverage is actually reduced and thus frequency counts are decreased. Subsequently, frequency-based solutions will fail to detect the increase in the Streaming API, and subsequently fail to detect the emerging event. The same phenomenon can also affect *standing keyword-based queries* (i.e., constantly streaming a small sample of tweets that contain specific keywords), which is the standard method of extracting Twitter content on behalf of specific

[1] See for example the official BBC Twitter account at http://twitter.com/#!/BBC

[2] http://en.wikipedia.org/wiki/Arab_Spring

[3] http://articles.cnn.com/2009-06-26/tech/michael.jackson.internet_1_google-trends-search-results-michael-jackson

[4] For the remainder of this article, will use the terms "event" and "topic" interchangeably.

[5] For a complete reference of options for streaming content from Twitter please refer to https://dev.twitter.com/docs/streaming-api/methods

Received June 20, 2014; revised October 28, 2014; accepted November 2, 2014

© 2015 ASIS&T • Published online 5 May 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.23465

entities by reputation monitoring systems. In contrast, sentiment-based event detection solutions could potentially circumvent this issue, as they would be able to detect an event that occurs in relation to a prespecified entity (e.g., company or product name) by detecting the sudden change of affective polarity that is associated with the entity, regardless of the decrease in data coming through the Streaming API.

The rest of the article is structured as follows. In the next section, we present an overview of the research for event detection. Subsequently, we present and analyze two state-of-the-art frequency-based solutions, that are used as baselines in our experimental setup, and describe our novel sentiment-based approach. We then present the Twitter data set that is used in our study along with the events that were used for evaluating all solutions, and the process that was employed for creating the golden standard. The section is followed by the presentation and analysis of a series of experimental results and we finish with conclusions and a discussion of future work.

Prior Work

Event detection has attracted significant attention for the better part of the last decade (Aiello et al., 2013; Allan, 2002). The series of MediaEval benchmarking activies that has been running since 2011 include a "social event detection" track with participants given a specific data set to analyze and specific events that have taken place within the timespan of the collection of the data set (Papadopoulos, Troncy, Mezaris, Huet, & Kompatsiaris, 2011). In contrast to the work presented here, the track is primarily focused on multimedia content, such as photo streams, rather than textual-based information.

Initial work on topic detection is based on standard latent Dirichlet allocation (LDA) models (Blei, Ng, & Jordan, 2003) extended to deal with dynamic data (Blei & Lafferty, 2006), and applied to large volumes of scientific publications. Nonetheless, as LDA is based on the co-occurrence of terms in texts, in order to extract the latent topics, it is unclear how well the approach would work on micro-blog posts, which are typically very short (140 characters long at maximum), use very idiosyncratic language and contain a lot of noise such as spam or meaningless "babbles" (Weng & Lee, 2011). Later, Platakis, Kotsakos, and Gunopulos (2009) attempt to discover *bursty* terms in the blogosphere (i.e., terms whose popularity is dramatically and unexpectedly increased) by applying Kleinberg's automaton technique (Kleinberg, 2003) on blog post titles. They additionally introduce a new representation of burstiness and employ Euclidean-based distance metrics to discover potential interterm correlations. They evaluate their solution by matching the results of the approach with real life events that occurred and report that they are able to successfully discover events such as the release of movies. Another approach that is based on the notion of burstiness is presented by Fung, Yu, Yu, and Lu (2005). Their approach uses the time information to determine a set of bursty features which may occur in different time windows and subsequently they detect such events based on the feature distributions.

Weng and Lee (2011) base their event detection solution on wavelet analysis. Their solution builds signals for each word by applying wavelet analysis on absolute token term frequencies. It then filters away the trivial words by looking at their corresponding signal autocorrelations and the remaining words are subsequently clustered to form events with a modularity-based graph partitioning technique. Culotta (2010) analyze Twitter messages by tracking the occurrence of flu-related keywords and are able to forecast future influenza rates with a high level of accuracy. Gruhl, Guha, Kumar, Novak, and Tomkins (2005) also track postings in blogs, media, and web pages using manually or automatically keyword queries, in order to predict spikes in book sales. More recently, Aiello et al. (2013) present a thorough comparison of different frequency-based solutions. Their analysis prominently includes the collapsed variational Bayesian inference algorithm (Teh, Newman, & Welling, 2007), which is based on LDA topic models (Blei et al., 2003) and the solution by Petrović, Osborne, and Lavrenko (2010) which is based on locality sensitive hashing (LSH), that is, locating near-duplicate documents in the Twitter stream. They also present a novel algorithm, named *BNgram*, which is able to capture events more robustly regardless of their time and the topical scope. We discuss the approach in more detail later, as it is used as example of frequency-based solution.

The work of Thelwall et al. (2011) is somewhat related to the approach presented here. They utilize sentiment analysis techniques to analyze popular events as they are discussed within Twitter, using SentiStrength (Thelwall, Buckley, Paltoglou, Cai, & Kappas, 2010). They conclude that popular events are typically associated with an increase in negative sentiment strength, regardless of whether the events themselves are positive in nature. In comparison to the approach presented in this article, they employ sentiment analysis only *after* a popular event has been identified, by the general increase of general term usage, while in our solution we explicitly use sentiment analysis techniques to discover such events. Additionally, their event detection approach is based on manually selected predefined keywords, an approach that would not be applicable in certain scenarios, as discussed earlier.

Event Detection in Twitter

Typically event detection algorithms are based on their capability to detect specific events, rather than every event that may have occurred in the timeframe in which they are examined (Aiello et al., 2013; Thelwall et al., 2011) as the latter analysis is impractical and unrealistic.

In this article, we examine whether a sentiment-based solution can provide an effective alternative to frequency-based solutions. As the basis for both analyses (i.e., frequency and sentiment based) we associate *every* hash-based word token (i.e., hashtag) extracted from the tweet stream with a time-series; of frequencies in the former case and sentiment in the latter in order to conduct the analysis. This approach is in contrast to limiting our data analysis to specific keywords (Thelwall et al., 2011). We also focus on the analysis of tweets on a daily basis, rather than exploiting shorter time periods (e.g., minutes or hours). We choose to focus on hashtags as they are often directly extracted from topics (e.g., *#olympics*) and typically strongly indicate the entity they are associated with. Applying the discussed solutions to any token is also feasible but would most likely require a preprocessing step of named-entity extraction (e.g., Aiello et al., 2013).

Frequency-Based Event Detection

Most event detection approaches are based on detecting sudden increases in the frequency of tokens. We compare our sentiment-based solution with two frequency-based approaches. They are described here.

Token spikes. We use the method originally presented by Thelwall, Prabowo, and Fairclough (2006) and also used in Thelwall et al. (2011), which has shown to be able to effectively capture events in various social media environments. The approach is based on *token spikes*, that is, increases of the absolute frequency of a token on a given day, compared to the average absolute frequency of the token in previous days.

More formally, a spike for token t is detected on day d if $freq_t(d) \geq threshold * avg_t(d-1, n)$, where:

$$avg_t(d,n) = \frac{1}{n+1}\sum_{i=d-n}^{d} freq_t(i) \qquad (1)$$

$freq_t(i)$ is the document frequency of token t (i.e., the number of tweets it appears in) on day i, n is the *past horizon* parameter (i.e., number of days before d) we consider to calculate averages, and $avg_t(d, n)$ is the average frequency of token t the n days before d. Lastly, *threshold* defines the value above which we consider a spike to have occurred. Thelwall et al. (2011) use a *threshold* value equal to 5. They also use all the previous days for extracting averages, that is, the value of n varies depending on which date is being analyzed. In our experiments, we use an initial value of $n = 3$, but also present results with different values, to explore the sensitivity of the algorithm to different settings.

BNgram. As previously discussed, Aiello et al. (2013) conduct a thorough analysis and comparison of event detection algorithms and present their own solution, named *BNgram*, that provides state-of-the-art effectiveness for a diversity of settings and parameters. It is based on the $df - idf_t$ metric, which is defined as:

$$df - idf_t(d,n) = \frac{freq_t(d)+1}{log\left(\frac{\sum_{i=d-n}^{d} freq_t(i)}{n}+1\right)+1} \qquad (2)$$

The main idea behind the approach is to detect events by comparing token frequencies at each day, with those of the preceding days. Although the underlying notion is the same as Thelwall et al. (2006), the estimation of token scores significantly differs Equation [2]. In addition, the method doesn't provide an explicit signal for detecting topics and events (e.g., based on the score of a token that increases above a specific threshold). Instead, it lists all tokens based on their $df - id$ scores for the time period that is under examination and considers the tokens with the highest scores to be representative of new, emerging events.

In their article, Aiello et al. (2013) also employ bigrams and trigrams, entity extraction, and clustering. Because our focus in this work is to provide a comparison between frequency- and sentiment-based event detection techniques, and we limit our analysis to hashtags, we avoid adding those elements in the implementation.

It should be noted that none of the frequency-based solutions considers the problem of token frequency normalization. That is, they are based on absolute counts of token counts rather than their relative frequency in the tweet stream (i.e., percentage of tweets that contain them). This can prove problematic, since the flow of content from Twitter (and the Streaming API in particular) can significantly vary from one day to the next. Such a phenomenon will become apparent when we discuss the specifics of the data set used in this study. Subsequently, a token's frequency may increase not because of the fact that a related event has taken place, but simply because there is a larger number of tweets in general, a subset of which contains the term. This subset can be of the same relative size as the previous days (e.g., 10% of tweets) but because of the larger number of tweets in general, a spike may be falsely detected. In contrast, as we will discuss, the sentiment-based approach is based on relative frequencies of tweets and is subsequently unaffected by this phenomenon.

Sentiment-Based Detection

The sentiment analysis event detection approach presented here is based on the assumption that a sudden change in the overall polarity that a keyword is typically associated with by concurrence, signifies an event relevant to that keyword. For this study, we use a simple ternary classification scheme, where every tweet can belong to one of three possible classes (i.e., $\{0, +1, -1\}$), where 0 signifies that the analyzed tweet contains only emotionally neutral, factual information, and $\{+1, -1\}$ mean that the tweet has either positive or negative content (e.g., funny, encouraging in the former case and argumentative in the latter). For example, consider the

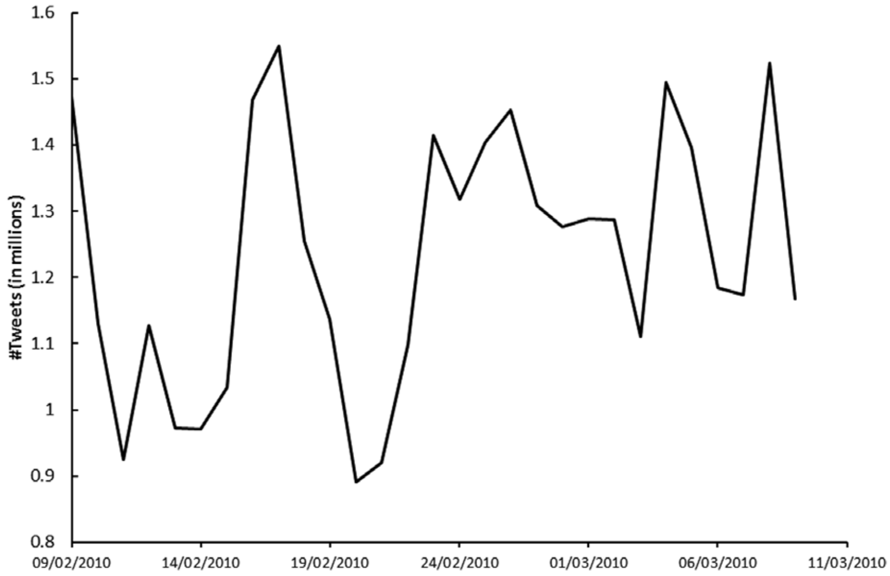

FIG. 1. Daily distribution of tweets contained in the data set.

Oscars ceremony; the approach would detect the particular event if a significant change of polarity in tweets that contain relevant keywords occur on the day of the event (e.g., the percentage of positive tweets that contain the hashtag *#oscars* the day before increases from 10% to 40% on the actual day of the event). We used the sentiment classifier by Paltoglou and Thelwall (2012), which has been shown to be effective in diverse social media environments, such as Twitter and Digg, without the need for training[6].

More concretely, based on the notation just introduced, we detect a *negative spike* for token *t* on day *d* if:

$$negFreq_t(d) \geq threshold * avgNegFreq_t(d-1, n) \quad (3)$$

where:

$$avgNegFreq_t(d, n) = \frac{1}{n+1}\sum_{i=d-n}^{d} negFreq_t(i) \quad (4)$$

Similarly, we detect a *positive spike* for token *t* on day *d* if:

$$posFreq_t(d) \geq threshold * avgPosFreq_t(d-1, n) \quad (5)$$

where:

$$avgPosFreq_t(d, n) = \frac{1}{n+1}\sum_{i=d-n}^{d} posFreq_t(i) \quad (6)$$

$negFreq_t(i)$ is the relative frequency[7] of negative tweets published on day *i* that contain term *t*. Respectively, $posFreq_t(i)$ is the relative frequency of positive tweets on day *i* that contain term *t*. Analogous to the frequency-based approach, we define an *emotional spike* as a sudden increase of the negativity or positivity that a word is associated with by concurrence. Our approach detects a *negative spike* if the relative number of negative posts containing a word in a day is increased above a threshold value in reference to the average relative number of negative posts that contain the term of the previous *n* days. The approach is similar for positive spikes.

For example, consider the token *#oscars*. If 20% (respect. 40%) of tweets that contain the term are positive (respect. negative) on average before a particular date, and on that date 50% (respect. 20%) of tweets are positive (respect. negative), then a positive spike will be detected, assuming a threshold value of 2. That is because, there was a twofold increase in the relative frequency of positive tweets that contain the term. As it can be observed, the absolute number of tweets is not considered.

Twitter Data set and Analysis

The Twitter data set that was used in this study contains a total of 35,750,897 tweets. It covers 1 full month of activity on the website, from February 9, 2010 until March 9, 2010, for an average of 1,232,789 tweets per day, limited to English tweets[8]. It was gathered using the Spinn3r API[9] which at that time allowed increased access to Twitter data for research purposes. Figure 1 demonstrates the daily distribution of tweets. It can be easily observed that the amount of content significantly varies from day to day. Indicatively, there were

[6] The software is freely available for research purposes at: http://www.cyberemotions.eu/

[7] As a reminder, we define *relative* frequency as the percentage of tweets rather than their absolute count.

[8] We intend to make the data set available to other researchers as the full list of urls of the posts that comprise it, making it possible for other researchers to recreate it.

[9] http://spinn3r.com/

TABLE 1. Events that we investigate in this article, ordered by chronological order.

# event	Event description	Date	Sample of relevant keywords
1	60*th* Berlin Film Festival (Opening)	2/11	#berlinale2010, #berlinale, #berlinfilmfestival, #berlinfestival
2	Vancouver Winter Olympics (Opening)	2/12	#vancouver, #olympics, #theolympics, #winterolympic, #2010winterolympics
3	Train collision in Hale	2/15	#buizingen, #traincollision, #trainaccidentinbelgium
4	BRIT Awards 2010	2/16	#brits, #brits2010
5	Tiger Woods apology	2/19	#tigerwood, #tigerspeaks, #tigersresponse
6	Littleton, Colorado school shooting	2/23	#littleton, #colorado, #shooting, #schoolshooting
7	7.0 earthquake strikes Japan	2/26	#earthquake, #earthq , #japanearthquake
8	USA census 2010	1/3	#uscensus, #uscensus2010, #2010census, #census2010
9	BBC to close BBC 6 Music	2/3	#savebbc6music, #savebbc, #savebbcasiannet, #savebbcradio6
10	82th Academy Awards Ceremony (Oscars)	3/7	#oscar, #academy, #awards, #oscars

Note. For an event to be considered successfully detected, one or more of the relevant keywords must be extracted within a pre-specified time-frame around the date of the event.

approximately. 1M tweets on February 15th, which increased to 1.4M tweets the following day, a 40% increase.

Table 1 presents the events that we focus on detecting in this study. The process of selecting the topics was similar to the approach adopted by Aiello et al. (2013); that is, from the potential list of topics that may have been selected, we identified events that were deemed significant by the fact that they gathered sufficient mainstream media attention. The time period covers a number of important events of world-wide interest, such as the 2010 Winter Olympics and the 82nd Academy Awards ceremony, as well as significant events of national-only interest, such as the BBC closing the 6 Music radio station, and a train collision in Hale, Belgium.

For each topic, the ground truth consists of an event id, a description, a date when it took place, and a set of keywords that are relevant to it. The hashtag-based keywords were created by a small group of experts who were given access to the data set and the list of topics and were asked to find hashtag-based keywords that strongly related to them. For example, for the USA census 2010 (event id: 9), the keywords *#uscensus* and *#uscensus*2010 were added. In all cases, the hashtag-based keywords were required to contain at least one of the topic description words or be uniquely connected to them (e.g., *#berlinale* for the Berlin Film Festival). This decision was made to avoid any obtuse keywords with the aim that an informed reader could easily deduct the event that has taken place from the hashtag-based keyword.

We use different daily sample *slices* of the data set to produce an analysis of the effectiveness of the event detection approaches in reference to the available data. Those include 1K, 10K, 100K, 500K, 750K, 1M, and *all* tweets from a day. This way, we can analyze the effectiveness of solutions having access to different orders of magnitude data sizes. This is the first time, to our knowledge, that event detection algorithms are tested with different samples of tweets to determine their effectiveness. It should be noted that for every slice, all algorithms analyzed the same sample of tweets. To avoid any sampling bias, we repeat the sampling process for each slice 10 times (apart from the *all* slice, since it contains all the data) and present results using the average.

To measure the effectiveness of all event detection algorithms, we use a variation of *topic recall* according to which a topic is considered to be successfully detected if *any* of the relevant keywords have been automatically identified at the date that the event took place. In contrast to the original definition by (Aiello et al., 2013) we do not require that all keywords be detected, because an informed reader can easily deduce the actual event by examining any of the keywords (e.g., #oscars2010) making the rest redundant. To limit potential noise from events that may have occurred at different times which may share keywords with the particular event, we limit the time-frame to the particular date that the event actually took place. We also use a looser ± 1 *topic recall*, according to which we extend the time-frame of detecting an event to ± 1 day in relation to the actual date. This is to take account of different time-zones throughout the world as the Twitter API often mispresents tweets from different time-zones. In addition, this definition will allow us to examine the after-effects of events, that is, whether users' interest in the events persists, increases or decreases after the event has been reported in mainstream media. For example, an algorithm would have considered to have detected the BRIT Awards 2010 ceremony (event id: 4) that took place on February 16 if any of the relevant keywords would have been identified on the dates February 15–17.

Results and Discussion

Table 2 presents the results using *all* tweets of the data set, with a past horizon *n* of 3 days to calculate averages and default algorithm parameters[10]. We examine the sensitivity of the sentiment-based solution to parameter variation in a following section.

It can easily be observed that some events are easier to detect than others. For example, all algorithms were able to detect the BRIT Awards ceremony and Tiger Wood's

[10] *threshold* = 5 for the *token spikes* (Thelwall et al., 2011), top 10 keywords extracted for BNgram (Aiello et al., 2013) and *threshold* = 3 for *sentiment spikes*.

DOI: 10.1002/asi

TABLE 2. Topic recall for all algorithms, using all the tweets in the data set.

Approach	Token spikes		BNgram		Negative spikes		Positive spikes	
# Event	Rec.	Rec. ± 1	Rec.	Rec. ± 1	Rec.	Rec. ± 1	Rec.	Rec. ± 1
1		√				√		√
2	√	√		√	√	√	√	√
3								
4	√	√	√	√	√	√	√	√
5	√	√	√	√	√	√	√	√
6		√				√		√
7	√	√				√		√
8					√		√	√
9	√	√	√		√	√	√	√
10	√	√		√	√	√	√	√
Avg. recall	60%	80%	20%	40%	60%	90%	60%	90%

Note. The “Rec.” columns measure strict topic recall, according to which, the exact date of the event must be extracted. The “Rec. ± 1” columns measure topic recall within a time-frame of 1 day of the exact date. The √ symbol denotes the successful detection of an event by the respective algorithm.

apology (event ids 4 and 5, respectively). In contrast, no algorithm was able to detect the train crash in Halle, Belgium despite the fact that it attracted significant worldwide media attention as the worst train accident over 50 years in the country. In fact, only a small amount of tweets that were published on the day of the accident contain the terms “halle” and “train” in the data set (54 out of approximately. 1M tweets), while in contrast 9.5K out of 1.1M tweets contain the terms “tiger” and “woods” the day that Tiger Woods made his public apology, in increase of two orders of magnitude. This phenomenon could signify a limitation of social-media based event detection algorithms as not every type of event attracts the same amount of social media coverage.

In terms of algorithm effectiveness, both the *token* and *sentiment* spikes algorithms seem to perform similarly. They all attain an average recall level of 60% using the strict recall metric, although they differ in the actual events that they were able to capture. For example, using this metric, the token spikes solution uniquely identified the earthquake in Japan while the sentiment spikes algorithms both identified the USA census. When using the looser *recall* ± 1 metric, their effectiveness increases to 80% for the token and spikes approach and 90% for the sentiment, as they are able to detect every event apart from the Halle train incident.

In contrast, the BNgram method seems to underperform, attaining much lower effectiveness levels. That is potentially due to the fact that in contrast to the other approaches that provide a clear signal of detecting an event when a token’s frequency or average sentiment value supersedes a threshold, it only ranks potential keywords in descending order of score and subsequently is very sensitive to the rank above which an event is detected. Note that this rank threshold is fundamentally different from the threshold value that the spike algorithms utilize, as in the latter case the threshold refers to observed frequency or sentiment increases, whereas in the former case it is a value that reflects the number of events that is expected to have been detected, something which can be difficult to predetermine or optimize.

Overall, the initial results indicate that sentiment spikes do provide an effective method of detecting events, performing equally or better than standard token-based solutions. There only seems to be minor differences in performance between the negative and the positive spikes approaches; a combination of both solutions could potentially provide a more robust solution, by providing multiple evidence of an event.

It is not surprising that both negative and positive spikes are being detected in some events, even when the actual event itself is *negative* in nature. That is because users either report the event using negative words or offer some type of support using positive language. For example, in reference to the Tiger Woods apology the following tweet is negative “RT @perezhilton: In case you didn’t know already . . . Tiger Woods is a f . . . g moron!” while the tweets “Is watching TIGER WOODS public Statement! Yes, in my opinion he is sincere!!!” and “I have just found @TigerWoods on twitter, Tiger, great speech, you’re a far better public speaker than our current President #TW2012” are positive. Similarly, in reference to the earthquake in Japan, the tweet “RT @nicholasscimeca: 7.0 earthquake in Japan? Mother Nature is pissed off. We need to stop f . . . g with her shit.” is negative, while “RT @ahkonlhamo: RT: @praybot RT @yingmin2 7.0 earthquake strikes off Okinawa. Please pray for those in Japan!!!” contains supportive, positive language.

Experiments with Different Sample Sizes

We now explore the effectiveness of the algorithms using different samples sizes of the data set. As previously mentioned, we present results with the algorithms having access to 1K, 10K, 100K, 500K, 750K, 1M tweets per day. For each

TABLE 3. Average topic recall for all algorithms, using different size samples of the data set.

Approach	Token spikes		BNgram		Negative spikes		Positive spikes	
Sample size	Rec.	Rec. ± 1	Rec.	Rec. ± 1	Rec.	Rec. ± 1	Rec.	Rec. ± 1
1K	32%	44%	4%	20%	8%	23%	0%	23%
10K	52%	65%	24%	39%	37%	54%	36%	52%
100K	58%	73%	29%	47%	43%	57%	47%	61%
500K	55%	81%	19%	39%	51%	72%	54%	70%
750K	55%	81%	20%	40%	55%	77%	57%	78%
1M	57%	81%	19%	39%	60%	86%	58%	82%
All	60%	80%	20%	40%	60%	90%	60%	90%

Note. The "Rec." columns measure strict topic recall while the "Rec. ± 1" columns measure average topic recall within a time-frame of 1 day of the exact date.

TABLE 4. Per query effectiveness of the token spikes approach for different sample sizes.

Metric	Recall					Recall ± 1				
# event	1K	10K	100K	500K	750K	1K	10K	100K	500K	750K
1	0%	0%	0%	0%	0%	0%	20%	50%	90%	90%
2	60%	70%	90%	100%	100%	100%	100%	100%	100%	100%
3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	90%	100%	100%	100%	100%	90%	100%	100%	100%	100%
5	70%	100%	100%	100%	100%	70%	100%	100%	100%	100%
6	0%	0%	0%	0%	0%	0%	10%	30%	80%	100%
7	10%	60%	50%	50%	50%	40%	100%	100%	100%	100%
8	0%	0%	40%	0%	0%	0%	20%	50%	40%	20%
9	20%	90%	100%	100%	100%	100%	100%	100%	100%	100%
10	70%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Avg.	32%	52%	58%	55%	55%	44%	65%	73%	81%	81%

Note. All entries are the averages over 10 different samples.

sample size, we present results using 10 different random samples of the data; all algorithms use the same sample for each run.

Table 3 presents the overall results. For reference, we also include the results using all the available data (from Table 2). Each entry in the table represents the average topic recall over 10 different samples and all 10 events. For example, the 32% recall for the token spikes approach using 1K samples signifies that over 10 different samples of 1,000 tweets each, the events were detected on average 32% of the time. For completeness, we report the results on individual queries over the different sample sizes in Tables 4–7, which we analyze subsequently.

The results in Table 3 indicate that, as expected, sample size is indeed an important factor that affects the performance of event detection algorithms. Most approaches seem to have a low detection rate when analyzing only 1K tweets per day, but that rate significantly increases when the sample sizes increase. In can also be observed that there are only incremental increases in effectiveness above a threshold number of tweets. That threshold number seems to vary depending on the employed algorithm; the token spikes solution has only minor increases in effectiveness after 100K tweets. The sentiment spikes solutions' performance keeps increasing until the 500K–750K tweets threshold. BNgram reaches its best performance at 100K tweets and then drops; the results may indicate that providing more data to the particular algorithm isn't always guaranteed to provide better performance, as the noise provided by the additional data seems to harm performance. Nonetheless, the overall results uniformly show that detection algorithms do not need to have access to massive amounts of data to perform adequately. Often, analyzing a smaller number of 100K tweets will provide a good enough result, after which the analysis of additional data may provide little or no benefits.

The results also show that, in contrast to our initial hypothesis, frequency-based solutions are able to function effectively even in environments where only a sample of social media data is available. Table 3 provides clear evidence that such solutions, exemplified by the token spikes approach, are able to reach a point after which only small increases in effectiveness take place much earlier that sentiment-based solutions. For example token spikes attains an average recall value of 55% at 100K tweets while the sentiment-based solutions reach the same level of recall at 750K and 500K for the negative and positive spikes solutions, respectively. Potentially, the utilization of a more

TABLE 5. Per query effectiveness of the BNgram approach for different sample sizes.

Metric	Recall					Recall ± 1				
# event	1K	10K	100K	500K	750K	1K	10K	100K	500K	750K
1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
2	0%	0%	0%	0%	0%	70%	100%	100%	100%	100%
3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	30%	100%	100%	100%	100%	30%	100%	100%	100%	100%
5	0%	60%	90%	90%	100%	0%	60%	90%	90%	100%
6	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
7	0%	0%	0%	0%	0%	0%	20%	80%	0%	0%
8	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
9	0%	10%	0%	0%	0%	0%	10%	0%	0%	0%
10	10%	70%	100%	0%	0%	100%	100%	100%	100%	100%
Avg.	4%	24%	29%	19%	20%	20%	39%	47%	39%	40%

Note. All entries are the averages over 10 different samples.

TABLE 6. Per query effectiveness of the negative spikes approach for different sample sizes.

Metric	Recall					Recall ± 1				
# event	1K	10K	100K	500K	750K	1K	10K	100K	500K	750K
1	0%	0%	0%	0%	0%	0%	0%	0%	50%	60%
2	20%	80%	90%	90%	80%	80%	100%	90%	100%	100%
3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	20%	100%	90%	100%	100%	20%	100%	100%	100%	100%
5	30%	80%	100%	100%	100%	40%	90%	100%	100%	100%
6	0%	0%	0%	0%	0%	0%	0%	30%	50%	80%
7	0%	20%	30%	10%	0%	0%	90%	100%	90%	60%
8	0%	0%	20%	40%	70%	0%	0%	20%	40%	70%
9	10%	20%	80%	100%	100%	0%	10%	20%	40%	70%
10	0%	70%	20%	70%	100%	80%	100%	40%	90%	100%
Avg.	8%	37%	43%	51%	55%	23%	54%	57%	72%	77%

Note. All entries are the averages over 10 different samples.

TABLE 7. Per query effectiveness of the positive spikes approach for different sample sizes.

Metric	Recall					Recall ± 1				
# event	1K	10K	100K	500K	750K	1K	10K	100K	500K	750K
1	0%	0%	0%	0%	0%	0%	0%	30%	30%	80%
2	0%	40%	40%	80%	90%	80%	80%	70%	100%	100%
3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	0%	100%	100%	100%	100%	20%	100%	100%	100%	100%
5	0%	100%	100%	100%	100%	40%	100%	100%	100%	100%
6	0%	0%	0%	0%	0%	0%	0%	0%	40%	80%
7	0%	40%	50%	20%	0%	0%	100%	90%	60%	40%
8	0%	0%	30%	80%	80%	0%	10%	40%	100%	80%
9	0%	40%	90%	100%	100%	10%	60%	100%	100%	100%
10	0%	40%	60%	60%	100%	80%	70%	80%	70%	100%
Avg.	0%	36%	47%	54%	57%	23%	52%	61%	70%	78%

Note. All entries are the averages over 10 different samples.

effective sentiment analysis algorithm would produce better results for the sentiment-based solutions, but sentiment analysis in social media in general is a very challenging and still a developing area of research (Paltoglou & Thelwall, 2012).

Still, the results do show that sentiment-based solutions are a possible solution for event detection and can still have unique advantages to frequency-based solutions. Those advantages are applicable in environments where data gathering is done via standing keyword-based queries,

TABLE 8. Topic recall for all algorithms with all the tweets in the data set, using two different past horizons of 2 and 5 days for extracting token frequency and sentiment averages.

	Past horizon = 2 days						Past horizon = 5 days					
	Token		Neg.		Pos.		Token		Neg.		Pos.	
# event	R.	R. ± 1	R.	R. ± 1	R.	R. ± 1	R.	R. ± 1	R.	R. ± 1	R.	R. ± 1
1		√		√		√	–	–	–	–	–	–
2	√	√	√	√	√	√	–	–	–	–	–	–
3												
4	√	√	√	√	√	√	√	√	√	√	√	√
5	√	√	√	√	√	√	√	√	√	√	√	√
6		√		√		√		√		√		√
7	√	√				√	√	√				√
8			√	√	√	√			√	√	√	√
9	√	√	√	√	√	√	√	√	√	√	√	√
10	√	√	√	√	√	√	√	√	√	√	√	√
Avg.	60%	80%	60%	80%	60%	90%	63%	75%	63%	75%	63%	88%

Note. The "R." columns measure strict topic recall, and "R. ± 1" columns measure topic recall within a time-frame of 1 day of the exact date. The "√" symbol denotes the successful detection of an event by the respective algorithm. The "–" symbol denotes that the particular event isn't eligible for detection as it falls within the training horizon.

where the rate of data streaming is stable, thus the applicability of frequency-based solutions is questionable and where the perception of the public towards an entity and the tracking of this perception over time are vital.

Analysis of Individual Topics Over Sample Sizes

Tables 4–7 report the effectiveness of the token spikes, BNgram, negative and positive spikes approaches for each individual topic over the different sample sizes. For readability reasons, we limit our analysis to 1K, 10K, 100K, 500K, and 750K sample sizes, since the 1M sample size performs very similar to the *all* data set (Table 2).

The results show that some topics are particularly easy to detect regardless of the sample size and the algorithm, while others are particularly challenging. For example, topics 4 and 5 (the BRIT Awards ceremony and the Tiger Woods apology) are always detected by most algorithms when using a sample size of 10K tweets. The same is also true to a lesser extend for event 10 (Oscars ceremony). In contrast, event 1 (the Berlin Festival) is never detected on the day of the event, but it is typically detected the following day. Of course, this result may be strongly correlated with the fact that the data set is strictly comprised of English tweets; a potential inclusion of German tweets may have provided a more accurate detection rate for the particular event. Similarly, event 6 (the Colorado school shooting) is never detected on the exact date of the event, but is detected the following day by most algorithms with sample sizes above 500K.

Overall, the experiments indicate that pop-culture events (e.g., music or film awards, public figure announcements) are easier to detect on the day they take place, as there often is a high level of expectation and media coverage before and during the event, while unpredictable negative events (such as accidents or natural disasters) usually take a while to percolate through social media and generally have much lower coverage. For example, only 65 tweets in the data set contain the word "littleton" and were published on the day of the event, in contrast to the approximately. 13K tweets containing the word "oscar" that were published the day of the Oscars ceremony.

Varying the Past Horizon

In experiments presented so far, we used a past horizon of 3 days to calculate token frequency and affective averages. In this section, we analyze and discuss the performance of algorithms when this horizon is varied. We present results using two other horizon values; 2 and 5 days, which, as the results show, provide sufficient evidence for useful conclusions. For brevity, we emit results for the BNgram approach, as token spikes have shown to be a better representative of frequency-based event detection algorithms. We use all the available tweets from the data set, therefore the results are comparable with the results presented in Table 2.

Results are presented in Table 8. Overall, they show that algorithms are not affected by the length of the training phase, and they perform similarly regardless of whether the training phase lasts 2, 3, or 5 days. That is, they are capable of detecting events even when token averages are calculated only on small number of days prior to events. The result may contradict initial preconceptions that a longer training period may have resulted in more robust token averages estimations, that is, more events would be successfully detected because the estimation of the average frequencies and sentiment would be based on longer periods of time. In contrast, the results show that for most scenarios, a smaller past horizon of 2 days can perform as well as a longer one of 5 days. The result is of significant value since maintaining

TABLE 9. Topic recall for sentiment-based algorithms with all the tweets in the data set, using three different threshold values (i.e., 2, 4 and 7) for detecting an affective spike.

Metric	Recall						Recall ± 1					
	Negative spikes			Positive spikes			Negative spikes			Positive spikes		
Threshold value	2	5	7	2	5	7	2	5	7	2	5	7
1							√	√	√	√	√	√
2	√	√	√	√	√	√	√	√	√	√	√	√
3												
4	√	√	√	√	√	√	√	√	√	√	√	√
5	√	√	√	√	√	√	√	√	√	√	√	√
6							√	√	√	√	√	√
7							√			√	√	√
8	√	√	√	√			√	√	√	√		
9	√	√	√	√	√	√	√	√	√	√	√	√
10	√	√	√	√	√	√	√	√	√	√	√	√
Avg.	60%	60%	60%	60%	50%	50%	90%	80%	80%	90%	80%	80%

Note. The "Recall" columns measure strict topic recall, and "Recall ± 1" columns measure topic recall within a time-frame of 1 day of the exact date. The "√" symbol denotes the successful detection of an event by the respective algorithm.

token frequencies for every day requires significant resources and a reduction of the days that need to be tracked from five to two can have significant positive results in terms of system memory management.

Sensitivity of Sentiment-Based Event Detection Algorithms to Threshold Values

All results presented so far are based on a threshold value of three for the sentiment-based event detection solutions (formula 3). That is, for example, a negative token spike is detected if the relative number of negative posts that contain the token in a day are more than three times the average relative number of negative posts that contain the token in the previous *n* days (where *n* is the past horizon parameter that we discussed in the previous section). In this section we investigate the sensitivity of the proposed algorithms to variations of this parameter.

Results are presented in Table 9. As we are using all the tweets in the data set, the results are comparable to those of Table 2. We experiment with three additional threshold values: two, five, and seven. Higher threshold values are expected to decrease the number of token spikes that are being detected.

Overall, the results indicate that both sentiment-based approaches are relatively robust to parameter variation, performing at a similar level regardless of the threshold value. The negative spikes approach performs slightly better, as it is able to retain 60% strict recall with higher parameter values of five and seven, in contrast to the positive solution that drops to 50%, because of the fact that at higher parameter values it fails to detect the US census 2010 (event id: 8).

A fact that isn't presented in the tables is that higher thresholds result in less keyword spikes being detected for each event. For example, when using a threshold value of two, there are four unique tokens that are being detected for the BRIT Awards event using the negative spikes approach, while using a value of five or seven results in only two unique tokens being detected. As all keywords are strongly indicative or uniquely associated with the events under study, the additional keywords don't affect the recall of the algorithms. This indicates that using higher thresholds may be advantageous as they decrease the number of affective token spikes that refer to the same event, thus reducing the amount of noncontributing keywords extracted by the algorithms.

Local Event Detection

In the previous sections we saw that local events, such as the train accident in Belgium (event id: 3), receive rather low social media attention and are particularly difficult to detect. In this section, we investigate this issue more fully by including five additional events in our analysis that have mainly a nation-wide interest. They are shown in Table 10.[11]

Table 11 presents the results of the event detection task for all approaches. Two things are easily observable; first, the BNgram method is rather inappropriate for detecting local news, as it failed to detect them in all cases. This issue is most likely due to the limitation of the approach of not providing a concrete signal for emerging events and relying on creating a list of keywords with decreasing values of *tf—idf*. It is assumed that should the interested party increase the list threshold to the top, for example, hundred detected keywords some of the events would be detected, but this type of optimization is beyond the scope of this article.

[11] Admittedly, event 12 may be considered to have a wider than national interest, as cricket is a popular sport in several countries, but we consider it here as it reflects the interests of a relatively small audience.

TABLE 10. National-based events used to measure the effectiveness of the event detection algorithms.

# event	Event description	Date	Sample of relevant keywords
11	Bakery blast attack in Pune, India	13/2	#pune, #puneblast, #blast
12	Plane crash in Austin, Texas	18/2	#austin, #atxplanecrash, #austinplanecrash
13	Sachin Tendulkar breaks cricket record	24/2	#sachin, #tendulkar, #sachiningod
14	Cyclone Xynthia hits France	28/2	#xynthia
15	First day of elections in Iraq	4/3	#iraqelections #iraq

TABLE 11. Topic recall for all algorithms, using all the tweets in the data set for the national-based events.

Approach	Token spikes		BNgram		Negative spikes		Positive spikes	
# event	Rec.	Rec. ± 1	Rec.	Rec. ± 1	Rec.	Rec. ± 1	Rec.	Rec. ± 1
11	√	√			√	√	√	√
12	√	√			√	√	√	√
13	√	√				√	√	√
14		√			√	√	√	√
15							√	√
Average	60%	80%	0%	0%	80%	100%	100%	100%

Note. The “Rec.” columns measure strict topic recall, according to which, the exact date of the event must be extracted. The “Rec. ± 1” columns measure topic recall within a time-frame of 1 day of the exact date. The √ symbol denotes the successful detection of an event by the respective algorithm.

Second, as previously discussed, some events are particularly difficult to detect, despite their significance and the level of attention they would be expected to attract. The Iraq elections, for example, although of great interest nationally are almost undetectable by most approaches with the exception of the positive approach. Concretely, there are only four tweets on March 4th (the first day of the elections) that contain the hashtag #iraqelections and only 33 that contain the hastag #iraq out of a total of approximately. 1.5M tweets that were published on the day. Still, a significant number of them contains positive, supportive language (e.g., “Tomorrow, Iraqis will have options and choices. Please make it #Iraq for Iraqis. #iraqelections”), which provides an indication why the positive approach was able to detect the particular event.

With regard to the rest of the approaches, the token and negative spikes approaches seem to perform similarly, both detecting events 11 and 12, whereas the former uniquely detects event 13 and the latter event 14. As previously, using the “recall ± 1” metric increases the effectiveness of all approaches, indicating that for local events in particular it is often the case the social media take 1–2 days to catch up with emerging news; as a consequence event detecting algorithms need at least that amount of time to be capable to extract such significant events. The phenomenon seems to also be reinforced by the results from Table 2, where it is typically local-focused events that are detected using “recall ± 1” (such as the Colorado school shooting or the Japan earthquake), while events with wider, global interest (such as the Oscars ceremony or the Olympic Games) are detected on the day of the event. Overall, it can be concluded that even though sentiment-based event detection solutions seem robust in reference to the social media attention events accrue, they still do require that the event gathers some attention in the medium in order to be able to detect it.

Conclusions

In this article, we investigated whether sentiment analysis techniques can be successfully utilized for event detection. The solutions proposed are based on detecting sudden changes in the overall negative or positive sentiment that keywords are associated with. We focused our analysis on the micro-blogging service Twitter, using a data set of over 32M posts that spans over a month.

Overall, the experimental results showed that sentiment-based solutions produce comparable performance with frequency-based approaches, being able to detect events on the day they occurred with the same level of recall as them, and being more effective in detecting them within a 1-day period. The results are very important as they indicate the potential of sentiment-based solutions for detecting events using social media. In a real-world situation where the sampling is based on standing keyword-based queries, such solutions can help interested parties track the general mood of the public in relation to specific entities and provide alerting mechanisms when spikes in negativity, indicative of an event that may need affirmative action, take place. In contrast to our initial hypothesis, we also demonstrated that frequency-based solutions are able to function effectively in environments where only a sample of social media content is available.

We also showed that the size of the sample used for event detection is a vital factor for any algorithm. Small sample sizes are sufficient for detecting popular-culture events by most algorithms, but some events need a significantly larger pool of posts to be detectable by either frequency- or sentiment-based solutions. In this aspect, the token spikes solution proved to be more effective, as it achieved a performance which is similar to using the full data set, when using only 100K tweets per day. In contrast, sentiment-based solutions require a larger sample, of approximately. 500K, to reach similar performance.

All algorithms proved very robust to changes in the parameter that defines how many days in the past are used to calculate token frequency or sentiment averages. We experimented with values of 2, 3, and 5 days and saw little difference in recall in all algorithms. These results indicate that even a parameter value of 2 days, which would provide important benefits in terms of system memory management, is sufficient for most algorithms to function effectively in most scenarios.

Lastly, an exploration of the threshold values that define the occurrence of a sentiment spike showed that both the negative and the positive approach provide a robust solution. Although higher values result in less keywords being detected, the event detection rate of algorithms remains unchanged, indicating that using higher values might be beneficial as it would reduce the number of unique tokens that refer to the same event being detected.

In the future, we intend to explore ways of combining the outputs from multiple event detection paradigms in building ensembles of algorithms, each based on different types of signals. Such a hybrid solution could combine both frequency- and sentiment-based solutions, in order to increase the coverage of the analysis. For example, if the outputs from both the token and negative spikes were combined when using a past horizon of 2 days (Table 8), the hybrid solution would be able to capture both events 7 and 8 (that were uniquely captured from each solution, respectively) and attain an increased recall rate. A thorough analysis of the capabilities and challenges of such a combination could provide significant advantages.

References

Aiello, L.M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., & Jaimes, A. (2013). Sensing trending topics in twitter. IEEE Transactions on Multimedia, 15(6), 1268–1282.

Allan, J. (Ed.). (2002). Topic detection and tracking: Event-based information organization. Norwell, MA: Kluwer Academic Publishers.

Blei, D.M., & Lafferty, J.D. (2006). Dynamic topic models. In Proceedings of the 23rd International Conference on Machine Learning (pp. 113–120). New York: ACM. doi:10.1145/1143844.1143859. Retrieved from http://doi.acm.org/10.1145/1143844.1143859

Blei, D.M., Ng, A.Y., & Jordan, M.I. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, 3, 993–1022. Retrieved from http://dl.acm.org/citation.cfm?id=944919.944937

Culotta, A. (2010). Detecting influenza outbreaks by analyzing twitter messages. CoRR, abs/1007.4748. (Informal publication).

Fung, G.P.C., Yu, J.X., Yu, P.S., & Lu, H. (2005). Parameter free bursty events detection in text streams. In Proceedings of the 31st International Conference on Very Large Data Bases (pp. 181–192). Trondheim, Norway: VLDB Endowment. Retrieved from http://dl.acm.org/citation.cfm?id=1083592.1083616

Gruhl, D., Guha, R., Kumar, R., Novak, J., & Tomkins, A. (2005). The predictive power of online chatter. In Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (pp. 78–87). New York: ACM.

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. Data Mining and Knowledge Discovery, 7(4), 373–397. doi:10.1023/A:1024940629314

Morstatter, F., Pfeffer, J., Liu, H., & Carley, K.M. (2013). Is the sample good enough? Comparing data from twitter's streaming api with twitter's firehose. In ICWSM (pp. 400–408). Boston, USA: The AAAI Press.

Paltoglou, G., & Thelwall, M. (2012). Twitter, myspace, digg: Unsupervised sentiment analysis in social media. ACM Transaction on Intelligent Systems and Technology, 3(4), 66:1–66:19.

Papadopoulos, S., Troncy, R., Mezaris, V., Huet, B., & Kompatsiaris, I. (2011). Social event detection at mediaeval 2011: Challenges, dataset and evaluation. In MediaEval. CEUR Proceedings Vol. 807: Pisa, Italy.

Petrović, S., Osborne, M., & Lavrenko, V. (2010). Streaming first story detection with application to twitter. In Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics (pp. 181–189). Stroudsburg, PA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1857999.1858020

Platakis, M., Kotsakos, D., & Gunopulos, D. (2009). Searching for events in the blogosphere. In Proc. of WWW'09 (pp. 1225–1226). Madrid: Spain.

Teh, Y.W., Newman, D., & Welling, M. (2007). A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In Advances in neural information processing systems (Vol. 19, pp. 1353–1360). MIT Press.

Thelwall, M., Prabowo, R., & Fairclough, R. (2006). Are raw rss feeds suitable for broad issue scanning? A science concern case study. Journal of the American Society for Information Science Technology, 57(12), 1644–1654.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment in short strength detection informal text. Journal of the American Society for Information Science Technology, 61(12), 2544–2558.

Thelwall, M., Buckley, K., & Paltoglou, G. (2011). Sentiment in twitter events. Journal of the American Society for Information Science Technology, 62, 406–418.

Weng, J., & Lee, B.-S. (2011). Event detection in twitter. In ICWSM (pp. 401–408). Barcelona, Spain: The AAAI Press, Menlo Park, California.