Event detection in Twitter: A machine-learning approach based on term pivoting

Florian Kunneman [a] Antal van den Bosch [a]

[a] *Centre for Language Studies, Radboud University*
P.O. Box 9103, NL-6500 HD Nijmegen, The Netherlands

Abstract

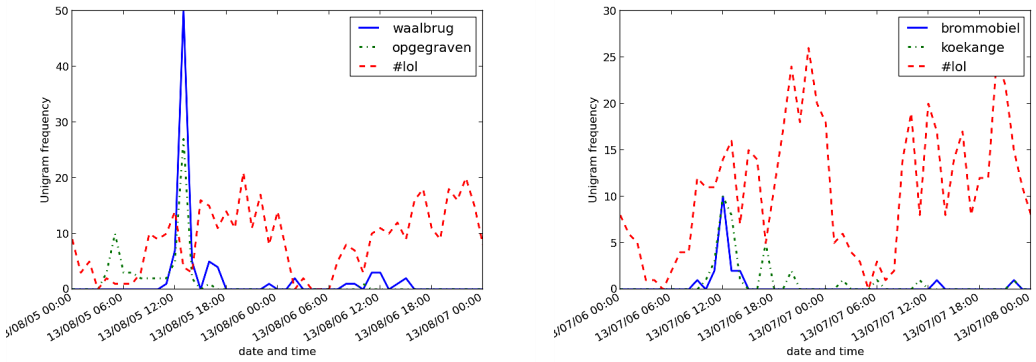
The large number of messages on Twitter posted each day provide rich insights into real-world events and public opinion. However, it is difficult to automatically distinguish tweets referring to such events from everyday chatter, and subsequently to distinguish significant events affecting many people from insignificant events. We apply a term-pivot approach to event detection from the Twitter stream. In order to filter out noisy and mundane events, we train a machine learning classifier on several rich features, and rank the events based on classifier confidence. After training and re-training the classifier using manually annotated data, we obtain an $F_{\beta=1}$ score of 0.79. However, a baseline that only takes into account the frequency of the tweets that refer to an event yields a better $F_{\beta=1}$ score of 0.86. We argue that performance is highly related to the definition of what makes a significant event, and that human understanding of this concept is not uniform.

1 Introduction

Microblogging platforms such as Twitter give users a voice to share ideas, opinions, and experiences with friends and the general public. Owing to the large user base on Twitter, the platform provides real-time information about what happens in the world. Detecting events and harvesting references to them from Twitter is therefore a highly valuable goal. However, this task is hampered by the nature and dynamics of Twitter. While news media select newsworthy items to write about, there is no such top-down selection process in the Twitter ecosystem. Events of public interest and mundane, insignificant events may both be characterized by bursty peaks in the usage of a set of terms in Twitter.

To give an impression of term burstiness in Twitter, consider the two examples in Figure 1. Example (a) displays the event of an excavation near the bridge 'Waalbrug' in Nijmegen, represented by a single joint rise and fall in the usage of the words 'waalbrug' and 'opgegraven' (Dutch for 'excavated') in Twitter. As a comparison, we also plot the frequency of the frequently used hashtag '#lol' in the same time window, which does not show any burstiness. It could be hypothesized that the first two terms both refer to an event, and possibly to the same event. Example (b) shows a similar pattern for the terms 'brommobiel' and 'koekange', peaking at about the same point in time, contrasted again with the non-bursty hashtag '#lol'. Without any additional knowledge, a system that leverages term burstiness might label the joint peaks in both examples as an event. However, further inspection shows that the peaks in example (b) denote a news report about a criminal act in the place of Koekange and an unrelated traffic accident with a scooter. A proper event detection system needs to filter out such insignificant events, possibly by taking into account additional features beyond burstiness.

The aim of this research is to expand existing work on detecting significant events on Twitter. We build on an approach proposed by [10]. They implement the *Twevent* approach to event detection in Twitter [7], and expand it by training a classifier on several features of an event to recognize significant events in contrast to mundane, insignificant events. We reproduce their experimentation and apply it to two months of Dutch tweets.



(a) (b)

Figure 1: Illustration of bursty and non-bursty term occurrences. Left: 'waalbrug' and 'opgegraven' (bursty) and '#lol' (non-bursty); right: 'brommobiel' and 'koekange' (bursty) and '#lol' (non-bursty).

2 Related Work

The detection of events in Twitter has been the goal of many studies. It is mainly approached as a clustering problem, with burstiness as the most important characteristic to detect an event. The most salient dichotomy among approaches is what [5] call *document-pivot clustering* and *term-pivot clustering*: burstiness is either measured at the level of tweets that share common terms, or at the level of single terms that display a joint burstiness over time. We provide an overview of the most important event detection systems, and summarize the performance on retrieving significant events reported by these studies.

2.1 Document-pivot clustering

The clustering of documents for the detection of events originates from the Topic Detection and Tracking (TDT) area of research [1]. Given a stream of news messages, any incoming message is linked to an existing event cluster or is the start of a new event cluster. [9] propose an adaptation of this approach to fast text streams such as Twitter. Incoming messages are either linked to an existing cluster, or grouped into a new one dependent on the distance to their nearest neighbour. Events are distinguished from other clusters based on the growth rate of a cluster. [9] obtain an average precision of 0.34 of retrieved event tweets versus tweets not related to an event, or spam. [8] reproduce the approach of [9], resulting in the retrieval of 1,340 events in 28 days of tweets, of which 382 (28%) are found to be significant.

Instead of clustering incoming tweets based on their raw content, alternative approaches focus on specific aspects of tweets that refer to future events. [11] state that important events on Twitter, in comparison to mundane events, have a common point in time to which multiple tweets refer explicitly. They extract events by clustering tweets that refer to the same point in time and mention the same entity. When ranking events based on the strength of the association between their date and entity, [11] obtained a P@100 (precision within the top-100 events) of 0.9 and a P@1000 of 0.52.

Yet another way to cluster tweets into events is to apply Latent Dirichlet Allocation (LDA) [2], by which individual words are linked to a topic based on their co-occurrence with other words. To detect bursty topics in Twitter, [4] build on the Twitter-tuned LDA implementation by [15], and expand it by adding topic distributions per time window and per user. Bursty topics are typically detected as a set of tweets from different users that contain similar words within a time window. A disadvantage of LDA is its dependence on parameter settings such as the number of topics, and the large number of sampling iterations that are required, leading to an extensively long duration for large sets of tweets. [4] set the number of topics to 30 in a period of 91 days, and obtained a precision of 0.76 for these topics (a precision@5 of 1.0).

2.2 Term-pivot clustering

[5] propose term-pivot (or feature-pivot) clustering as an alternative to document-pivot clustering for event detection from a news stream. Its two main advantages are the independence from parameter settings, and the event summary that is readily given by clustered terms. The first effective application of term-pivot clustering to event detection on Twitter is proposed by [14], who capture the burstiness of words by approaching them as signals and applying wavelet analysis to them. They obtain a precision of 0.76 for 21 events retrieved in a month of tweets from a Singapore user base.

[7] argue that multi-word segments or word n-grams, rather than single words, are beneficial both for the interpretation of an event and the detection of significant events. At the core of their *Twevent* system is the extraction of meaningful n-grams from tweets. N-grams are scored by their burstiness, and bursty n-grams are clustered into candidate events. The significance of a candidate event is dependent on the *Newsworthiness* of the individual n-grams, formulated as the combined chance of any n-gram subphrase to occur as an anchor text in Wikipedia, and the mutual similarity scores between the n-grams. [7] obtained a precision of 0.86 for 101 detected events on the same dataset as [14].

For the works discussed above, event significance is scored by an intuitive measure, such as the number of cluster terms [14] or the growth rate of a cluster [9]. Aiming to improve over these simple estimations of event significance, [10] apply *Twevent* to 15 days of English tweets and annotated the 4,249 resulting clusters as 'True news event' or 'False news event'. The clusters are linked to 15 rich features presumed to be indicative of their significance (these features are described in more details in Section 3.3.1). A classifier is trained and tested through 10-fold cross validation on all event clusters, resulting in a precision of 0.84 on 146 retrieved events, compared to 0.76 on 107 events by the original *Twevent* system.

In the study described here we adopt the approach by [10]. Where [10] build on the framework of *Twevent* to form clusters of segments, we base this clustering on unigrams rather than on segments. The rationale behind this is that in Dutch, the language we work with, word formation is characterized by compounding, which means that Dutch unigrams to a certain extent capture the same information as English bigrams. Compare, for instance, 'home owner' to 'huizenbezitter'.

As a definition of what makes an event significant, we follow the definition given by [8]: 'Something is significant if it may be discussed in the media.' As a proxy, we borrow the idea of [7] to include the presence of a certain name or concept as an article on Wikipedia as a weight in determining the significance of the candidate cluster of terms.

3 Experimental Set-up

3.1 Data

We collected two months of Dutch tweets by means of `twiqs.nl`, an archive of Dutch tweet IDs from December 2010 onwards [12]. The tweets in `twiqs.nl` are collected continuously from the Twitter API on the basis of a seed list of Dutch words and a list of the most active Dutch users. The harvesting is limited.[1] We collected the available tweets from 2013/06/22 until 2013/08/22, and filtered out non-Dutch tweets according to the language identification offered by `twiqs.nl`, resulting in a set of 65.02 million tweets.

3.2 Event detection

Our event detection approach takes the following steps.

3.2.1 Unigram selection by burstiness

To select candidate unigrams we first tokenize the tweets with ucto,[2] remove punctuation and user names, and lowercase the remaining words. Additionally, we remove stop words from each tweet. For each unigram we generated a time sequence of the tweets that contain the unigram. Following [7] we set the window size for this sequence to 24 hours, focusing on events that occur within a day.

[1] `twiqs.nl` harvests an estimated half of all Dutch tweets.

[2] `http://ilk.uvt.nl/ucto`

Given a day-by-day sequence of counts for a unigram, we score its burstiness per day by applying the state automaton approach to burstiness detection [6]. Each day a unigram can take on a bursty or normal state. The most likely sequence of states for a unigram can be uncovered by applying a Hidden Markov Model on the observed probability at each stage and the transition probability from state to state. We base the modeling of these two probabilities on the implementation by [4]. The observed probability of a count is based on a Poisson distribution for each state, which is defined as follows:

$$p(f_{ut} \mid v_t = l) = \frac{e^{-\mu_l} \mu_l^{f_{ut}}}{f_{ut}!} \tag{1}$$

Where f_{ut} is the frequency f of unigram u for time window t, l is either 0 or 1, and the normal and bursty states are denoted by μ_0 and μ_1, respectively. Following [4], we set μ_0 to the average count of a unigram over time and we set $\mu_1 = 3\mu_0$, i.e. an observed frequency has a higher probability to represent a bursty state when it approximates three times the average count. Also following [4] we set the transition probability σ_0 to 0.9 and σ_1 to 0.6, implying that a transition from a normal state to a bursty state is not very likely with a chance of 0.1. The chance that a bursty state reverts to a normal state is higher, with 0.4.

We use the Viterbi algorithm to dynamically find the bursty states for each unigram, and discard the unigrams without a bursty state as candidates. In our data set of 61 days, the method identifies $253,472$ bursty unigrams, with an average of $4,088$ per day ($\sigma = 703$).

3.2.2 Unigram similarity

To cluster unigrams into event clusters, we adopt the approach by [7]. For each day in our dataset, the similarity between all pairs of bursty unigrams is calculated and clusters are formed based on this similarity graph. To calculate the similarity, each time window t is divided into M sub-time-windows. Following [7] we set the size of M to 12 (i.e. two hours per sub-time-window). The similarity between any pair of unigrams u_a and u_b on a day is calculated as follows:

$$sim(u_a, u_b) = \sum_{m=1}^{M} w_t(u_a, m) w_t(u_b, m) sim(T_t(u_a, m), T_t(u_b, m)) \tag{2}$$

The sub-time-window similarity between unigrams is computed by collecting the tweets in which the unigrams are mentioned, and generating two pseudo-documents containing all concatenated tweets in which one or the other unigram occurs. Terms in these documents are weighted by $tf - idf$, and the cosine distance between the two pseudo documents is calculated as the similarity score between the two unigrams. This calculation favors pairs of unigrams that are mentioned with comparable content and that are most bursty in the same sub-window. Furthermore, it considers the similarity between tweets rather than the co-occurence of unigrams, which is reasonable given the shortness of tweets.

3.2.3 Term clustering

Given the similarity graphs of bursty unigrams per day that result from the previous step, unigrams are clustered into event clusters. Following [7], we apply Jarvis-Patrick clustering. This algorithm has two parameters, k and l. For any two unigrams to be clustered together, they have to occur in each others k-nearest neighbours and they have to share at least l common neighbours in their k-nearest neighbors. Advantages of this algorithm are its limited computational cost and the fact that the number of topics does not have to be defined.

[7] found that the l parameter is too restrictive for this task. Following them, we only took into account the k parameter and set $k = 3$, linking unigrams if they occur in each other's top-3 most similar unigrams. Unigrams that were not linked to any other unigram were discarded. As a result, we retrieved a total of $33,452$ event clusters from the 61 days of bursty unigrams (548 on average per day).

3.3 Event significance classification

Event significance classification can be seen as what [10] call 'event filtering'. The events that result from clustering are sorted into significant and insignificant events. We apply the same approach to

event filtering as [10]: describing event clusters by rich features and training a classifier to distinguish significant from insignificant events.

3.3.1 Features

In their research, [10] include 15 rich features. Most of the features that we use are adopted from [10]. We describe the features below, and make a distinction between cluster features and tweet features: respectively the characteristics of the unigrams that describe a cluster and the characteristics of the tweets in which the unigrams of a cluster occur on the day of their burstiness (referred to as event tweets). For each of the 15 features, we give a full name and an abbreviation between brackets, which will henceforth be used to refer to the feature.

Cluster features

- Unigrams (UNI) - the number of unigrams in the event cluster. Arguably, a cluster which is described by many unigrams is not likely to represent a coherent, significant event.
- Edges (EDGE) - the average number of clustering edges between the unigrams in the event cluster. This feature describes the density of a cluster.
- Similarity (SIM) - the average similarity score, as described in section 3.2.2, between unigrams in the event cluster.
- Burstiness (BST) - the average burstiness of unigrams in the event cluster, adopted from the bursty probability calculation in [7]. This probablity is based on the expected frequency $E[u|t]$ of a unigram u in a time window t, given its Gaussian distribution:

$$E[s|t] = N_t P_s = N_t * \frac{1}{l} \sum_{t=1}^{L} \frac{f_{u,t}}{N_t} \tag{3}$$

  Here, N_t is the number of tweets during day t, L is the number of time windows containing u, and $f_{u,t}$ is the frequency of u in time window t. Given $E[s|t]$, the bursty probability $P_b(s,t)$ is calculated as follows:

$$P_b(s,t) = S(10 * \frac{f_{s,t} - (E[s|t] + \sigma[s|t])}{\sigma[s|t]}) \tag{4}$$

  S is the sigmoid function and $\sigma[s|t] = \sqrt{N_t P_s (1 - P_s}$, the standard deviation of the Gaussian distribution.
- Newsworthiness (WIKI) - the average newsworthiness of unigrams, which is operationalized in [7] as the ratio by which terms that are (in) the title of a Wikipedia page are referred to from other pages from anchored links. Terms that have a high probability to be used as anchor to their page are believed to be more newsworthy. To calculate the newsworthiness score for all bursty terms, we downloaded a dump of the Dutch Wikipedia pages from November 14th 2013 (the closest date after our data set).[3]

Tweet features

- Document Frequency (DF) - the relative frequency of the event tweets, calculated as the number of event tweets on the given day divided by the total number of tweets on that day.
- User Document Frequency (UDF) - the relative number of different users that refer to the event, calculated as the number of users that posted one of the event tweets, divided by the total number of event tweets.
- Hashtags (HT) - the average number of hashtags (#) per event tweet
- URLs (URL) - the percentage of event tweets that contain a URL (any token starting with ('http://')
- Replies (REP) - the percentage of event tweets that start with a username (tokens that start with a '@'), which is typical of a reply.

[3] http://dumps.wikimedia.org/nlwiki/20131114/

- Mentions (MEN) - the percentage of event tweets that contain a mention of a username, on any position other than the start of a tweet.
- Cohesiveness (CHS) - the average number of unigrams in tweets. If the event tweets contain two or more of the clustered unigrams, they are more likely to refer to a cohesive event.
- Informativeness (INF) - the relative number of different words in the event tweets. Spam messages are often characterized by a narrow vocabulary, while events that arouse the attention of a lot of people might be referred to with a bigger variation of words.

3.3.2 Classification

While [10] annotate all 4,249 event clusters retrieved by the *Twevent* approach from their data set, we did not annotate all 33,452 event clusters retrieved from our data set. Instead we selected a subset of the data. To make sure we had enough significant events in this subset, we trained a classifier on 350 labeled event clusters on the first two days in our data set and applied it to the remaining days. The 1000 events of which the classifier was most confident were used as data set for our experimentation.

As classifier we made use of the SNoW implementation of Winnow [3][4]. This algorithm is known to offer state-of-the-art results in text classification, and outputs a per-class confidence score by which instances could be ranked. To tune the different parameters of Winnow (α, β, $\theta+$, $\theta-$, the number of iterations and the thick seperator), we applied a heuristic hyperparameter optimization scheme that makes use of wrapped progressive sampling on training data [13].

To obtain labeled data for the preliminary classification we ranked the event clusters in the first two days based on the average similarity score of their unigrams. One of the authors annotated the top 350 of these events as significant or not, resulting in 153 events labeled as significant and 197 events labeled as insignificant. The classifier was trained on these 350 labeled events and was applied to all events in the remaining days in the data set. The 1,000 events that were most confidently scored as significant by the classifier were used in our main experimentation.

To obtain trustworthy labels for the 1,000 events we asked 8 annotators to each label 250 events as significant or not. The data was split in a way that each event was annotated by two annotators, with 8 unique annotator pairs (125 events per pair). We presented them with a list of events represented by a date, the event unigrams, and a sample of 10 of the event tweets. In our explanation of the task, we gave them the definition of a significant event that we specified in section 2.2, as well as a few examples of typically significant and insignificant event clusters. The task was to annotate each event as either significant, insignificant, or doubtful. We additionally asked the annotators to indicate if the event was a social event, which we planned to use for additional research.

354 of the 1,000 event clusters were indicated by both annotators as significant, 723 were annotated as significant by at least one of the two annotators and 277 events were annotated by both annotators as insignificant. The mean inter-annotator agreement was fair ($\kappa = 0.25$, with a standard deviation of 0.11).

3.4 Evaluation

Given the 1,000 annotated event clusters, we evaluated classification performance by 10-fold cross-validation. We apply classification with a strict and lax labeling. For strict labeling, only events that were indicated as significant by two annotators are labeled as significant, while for the lax labeling, events that were annotated by one as significant are seen as significant. To score the performance, we calculate the precision, recall and F1 scores for the retrieval of significant events. As baselines we ran the classifier separately on the intuitively most effective features for significant event classification: burstiness (BST), the number of tweets mentioning the event (DF), and the similarity between unigrams (SIM).

4 Results

The results are given in Table 1. Both in the strict and the lax setting the classifier that bases its judgements on all feature values yields a worse performance than one of the classifiers based on a single

[4] http://cogcomp.cs.illinois.edu/page/software_view/SNoW

feature. In the strict setting, the relative document frequency is sufficient, while for the lax setting the term burstiness leads to a peak performance of .94.

	Strict			Lax		
	Precision	Recall	F1	Precision	Recall	F1
DF	.80	.95	**.86**	.73	.99	.84
SIM	.54	.93	.68	.84	.95	.89
BST	.57	.84	.68	.93	.94	**.94**
All	.76	.90	.79	.91	.93	.92

Table 1: Results for the classification of events as significant in the strict and lax setting, by performing classification based on a single feature (DF, SIM or BST) and based on all 13 features.

In Table 2 we show the five events that were most confidently ranked as significant by the classifier that uses all features. Three of the events are arguable significant for a large number of people: the football match 'Spanje-Italie', a goal of Clarence Seedorf, and a reference to the television program 'Miracle Run'. The next event is of a personal nature (school performance), while the final case is only arguably newsworthy (breeding insects for improving the environment).

date	event terms	example tweet
27-06-2013	spanje-italie, italie	spanje-italie kijken ik ben voor itali
15-07-2013	botafogo clarence	RT @433NL VIDEO Oud maar nog steeds gedreven Clarence Seedorf 37 scoorde vanavond een heerlijke goal voor Botafogo http://t.co/Wj7hp
15-07-2013	efron zac miracle	@BBergstra op rtl 8 Miracle run Gaat over een autistische tweeling met Zac Efron x
03-07-2013	bevorderd gymnasium	Bevorderd naar gymnasium 3 :-)
04-07-2013	milieuproblemen kweken	Insecten kweken als op lossing voor voedsel en milieuproblemen http://t.co/cRTFvL7ToT

Table 2: events classified as significant most confidently based on all 13 features in a 10-fold cross-validation setting.

5 Conclusion and discussion

We reproduced the term-pivot approach to event detection proposed by [7] and applied it to two months of Dutch tweets. In line with [10] we annotated the resulting events on their significance and trained a machine learning classifier based on 13 features. We found that the relative frequency by which an event is mentioned provides a sufficient cue to recognize significant events as opposed to feeding the classifier all 15 features, yielding $F_{\beta=1}$ scores of 0.86 and 0.79, respectively.

Our system obtains precision values that are similar to the ones reported by [10] (around 0.80), while our recall values are much higher. An explanation is that [10] train and test on a much larger set of 4,249 events with a smaller fraction of significant events, making the task more challenging. Furthermore, we train and test on the already ranked output of our classifier. As [10] we find that the number of event tweets and the number of URLs in these tweets are important features to recognize significant events. On the other hand, user document frequency (UDF), newsworthiness (WIKI) and similarity values (SIM), reported by them as useful features, did not have a big influence on the classifier performance in our experiment.

In our experiment, two annotators labeled each event. The low agreement value ($\kappa = 0.25$) shows that it is difficult even for humans to decide whether an event is significant or not. We found that it is not trivial to provide the annotators with an unambiguous definition of what makes a significant event. In future work we will attempt to develop a sharper definition.

Acknowledgements

This research was funded by the Dutch national program COMMIT. We thank Erik Tjong Kim Sang for the development and support of the `http://twiqs.nl` service.

References

[1] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 37–45, New York, NY, USA, 1998. ACM.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] A. Carlson, C. Cumby, J. Rosen, and D. Roth. The snow learning architecture. Technical report, University of Illinois at Urbana-Champaign, 1999.

[4] Q. Diao, J. Jiang, F. Zhu, and E.-P. Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 536–544. Association for Computational Linguistics, 2012.

[5] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment, 2005.

[6] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[7] C. Li, A. Sun, and A. Datta. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 155–164. ACM, 2012.

[8] A. J. McMinn, Y. Moshfeghi, and J. M. Jose. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 409–418. ACM, 2013.

[9] S. Petrović, M. Osborne, and V. Lavrenko. 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*, pages 181–189. Association for Computational Linguistics, 2010.

[10] Y. Qin, Y. Zhang, M. Zhang, and D. Zheng. Feature-rich segment-based news event detection on twitter. In *International Joint Conference on Natural Language Processing*, pages 302–310, 2013.

[11] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 1104–1112, New York, NY, USA, 2012. ACM.

[12] E. Tjong Kim Sang and A. van den Bosch. Dealing with big data: The case of twitter. *Computational Linguistics in the Netherlands Journal*, 3:121–134, 12/2013 2013.

[13] A. Van den Bosch. Wrapped progressive sampling search for optimizing learning algorithm parameters. In *Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence*, pages 219–226, 2004.

[14] J. Weng and B.-S. Lee. Event detection in twitter. In *Proceedings of the AAAI conference on weblogs and social media (ICWSM-11)*, pages 401–408, 2011.

[15] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011.