Erschienen in: Advances in Intelligent Data Analysis XV : 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings / Boström, Henrik et al. (Hrsg.). - Cham : Springer, 2016. - (Lecture Notes in Computer Science ; 9897). - S. 368-380. - ISSN 0302-9743. - eISSN 1611-3349. - ISBN 978-3-319-46348-3
https://dx.doi.org/10.1007/978-3-319-46349-0_32

Stability Evaluation of Event Detection Techniques for Twitter

Andreas Weiler[1], Joeran Beel[2(✉)], Bela Gipp[1], and Michael Grossniklaus[1]

[1] Department of Computer and Information Science,
University of Konstanz, 78457 Konstanz, Germany
{andreas.weiler,bela.gipp,michael.grossniklaus}@uni-konstanz.de
[2] Digital Contents and Media Sciences Research Division,
National Institute of Informatics (NII), Tokyo 101-8430, Japan
beel@nii.ac.jp

Abstract. Twitter continues to gain popularity as a source of up-to-date news and information. As a result, numerous event detection techniques have been proposed to cope with the steadily increasing rate and volume of social media data streams. Although most of these works conduct some evaluation of the proposed technique, comparing their effectiveness is a challenging task. In this paper, we examine the challenges to reproducing evaluation results for event detection techniques. We apply several event detection techniques and vary four parameters, namely time window (15 vs. 30 vs. 60 mins), stopwords (include vs. exclude), retweets (include vs. exclude), and the number of terms that define an event (1...5 terms). Our experiments use real-world Twitter streaming data and show that varying these parameters alone significantly influences the outcomes of the event detection techniques, sometimes in unforeseen ways. We conclude that even minor variations in event detection techniques may lead to major difficulties in reproducing experiments.

1 Introduction

The continuous success of Twitter and its freely available data stream have fostered many research efforts specialized on social media data. In this area of research, event detection is one of the most popular topics. In general, all event-detection approaches have in common that they attempt to detect patterns that differ from the normal behavior of the data stream. However, there are different types of techniques that can be used for this task. For example, Weng *et al.* [29] and Cordeiro [9] use techniques that are based on wavelet transformation to detect the events. Other works, such as Alvanaki *et al.* [2] or Mathioudakis and Koudas [16], use statistical models to detect significant abnormalities.

A major challenge in event-detection research is reproducibility. Reproducibility describes the case in which the outcome of two experiments allows drawing the same conclusions [4]. For instance, if an experiment shows that Algorithm A has faster run-times than Algorithm B, the conclusion might be

that Algorithm A outperforms Algorithm B. This research would be considered reproducible if a similar experiment also leads to results that support the conclusion that Algorithm A outperforms Algorithm B.

Reproducibility is affected by three factors, namely the similarity of scenarios, algorithms, and evaluation techniques [4]. If two experiments use the same algorithms, in the same scenario and apply the same evaluation techniques, then one would expect the outcome of the experiments to be the same. However, algorithms, scenarios and evaluation techniques typically differ somewhat between two experiments. If these differences are sufficiently small, one would nevertheless expect the outcome of the experiments to be at least similar and to support the same conclusions.

Our previous research in the field of recommender systems showed that minor differences in the experimental setup can at times lead to significant differences in the outcomes of two experiments. In one experiment to assess the effectiveness of a recommendation approach, removing stopwords increased recommendation effectiveness by 50 % [6]. In another experiment, effectiveness was almost the same [5]. Similarly, Lu *et al.* [14] found that sometimes terms from an article's abstract performed better than terms from the article's body, but in other cases they observed the opposite. Zarrinkalam and Kahani [30] found that terms from the title and abstract were most effective in some cases, while in other experiments terms from the title, abstract, and citation context were most effective. Bethard and Jurafsky [7] reported that using citation counts in the recommendation process strongly increased the effectiveness of their recommendation approach, while He *et al.* [12] reported that citation counts slightly increased the effectiveness of their approach. In all these examples, the changes in the algorithms, scenarios, and evaluation methods were minor. Nevertheless, even minor changes led to significantly different outcomes of the experiments, meaning that many research results in the recommender system community must be considered as not reproducible.

In the research community that studies event detection in social media data streams, reproducibility has received little attention to date. Based on our previous research and experience in the area of recommender systems for scientific publications, we believe that research on event detection techniques must place more emphasis on the issue of reproducibility. Currently, many evaluations of event detection appear to be non-reproducible. Weiler [24] lists the evaluation methods of a collection of 42 research works on event detection. Half of these evaluations is based on case or user studies. Reproducing these studies can already be challenging due to the inherent human element. Also problematic is the use of different data sets, which makes it hard or even impossible to reproduce the results of an experiment. Often the data sets used are heavily pre-filtered for users and/or regions or obtained by applying keyword filters. To address this issue, some works attempt to create and provide labelled reference data sets to evaluate event detection techniques. For example, McCreadie *et al.* [18] created a set of approximately 16 million tweets together with a list of 49 reference topics for a two-week period. However, since the corpus focuses on ad-hoc retrieval

tasks and no description is given of how the topics were created, this reference data set is ill-suited for the evaluation of event detection techniques. Further reference data sets are proposed by Becker *et al.* [3], Petrović *et al.* [23], and McMinn *et al.* [19]. All of these corpora suffer from the shortcoming that the contained tweets need to be crawled. In the case of Twitter, crawling is a challenging task. With limited requests to the API it is almost impossible to retrieve all the tweets in a reasonable time frame. Also it is possible that a certain number of tweets are no longer available and therefore the final crawled corpora is not complete, which again limits the reproducibility of experimental results.

Based on a literature review of existing research, it can be observed that the terms "reproducibility" or "stability" are never mentioned as evaluation measures. Therefore, our research objective is to study the stability of event detection techniques as a necessary pre-condition for the reproducibility of event detection research. In the long run, the effect of all three factors (changes in algorithms, scenarios, and evaluation methods) need to be researched. However, for now, we focus on the first factor, *i.e.*, the effect of minor variations in event-detection algorithms. The research question of this paper is therefore: "How do minor changes in event detection techniques affect the reproducibility of experimental results?"

2 Methodology

To assess the reproducibility of experiments conducted with state-of-the-art event detection techniques, we study the stability of the obtained results w.r.t. slight variations in the parameter settings of these techniques. The studied event detection techniques all consist of a pre-processing, event detection, and event construction phase. For the evaluations presented in this paper, we varied parameters that affect the pre-processing and event detection. For the pre-processing phase, we conducted two experiments that respectively omitted the operators to suppress retweets and stopwords. In the event detection phase, we varied the size of the time-based window that is processed by the techniques. Based on these parameter variations, we studied the following configurations.

- 1 h windows with stopwords vs. without stopwords (pre-processing)
- 1 h windows with retweets vs. without retweets (pre-processing)
- 15 min vs. 30 min vs. 1 h windows (event detection)

For each of these configurations, we study the stability of the task-based and run-time performance results. In terms of task-based performance, we compare the results of a technique in one configuration to the results of the same technique in a different configuration. As all techniques report events as a set of five terms, we measure on how many terms in the two result lists overlap. In terms of run-time performance, we analyze how the different configurations influence the throughput (tweets/sec) of a technique. The rationale behind these experiments is that in order to be reproducible, small changes in the parameters should not drastically change the detected events. In other words, the more diverse the

detected events were, the less stable the algorithms are, and hence, the less likely it would be to reproduce the results obtained in one experiment.

2.1 Experimental Setup

The data sets used in our evaluation consist of 10 % of the public live stream of Twitter for three different days. Using the Twitter Streaming API[1] with the so-called "Gardenhose" access level, we collected data for the randomly chosen days of 15th April 2013, 13th March 2013, and 8th July 2014. On average, the data sets contain a total of 20 million English tweets per day and an average of 850,000 tweets per hour.

All experiments were conducted on server-grade hardware with 1 Intel Xeon E5 processor at 3.5 GHz with 6 cores and 64 GB of main memory, running Oracle Java 1.8.0_40 (64-bit). Regardless of the available physical memory, the *–Xmx* flag of the Java Virtual Machine (JVM) was used to limit the maximum memory to 24 GB.

2.2 Event Detection Techniques

The studied techniques were all realized as query plans (*cf.* Fig. 1) in the *Niagarino* data stream management system [27]. The operators with a dashed frame are the components that are modified in our experiments. The implementations and parameters of the first three techniques *EDCoW* [29], *WATIS* [9], and *Shifty* [25] have already been described in our previous work on evaluating event detection techniques [27].

In this paper, we additionally study the *LLH* and *enBlogue* (*ENB*) event detection techniques. *LLH* is a reimplementation of Weiler *et al.* [28]. In a first step, the technique aggregates and groups the distinct terms by their counts. Then the log-likelihood ratio operator collects n values per term as input signal. For the calculation of the log-likelihood ratio at least two windows need to be analyzed by the operator. After the analysis of two windows the log-likelihood ratio between all terms in the current window is calculated against the past. Events are reported by selecting the top N terms with the highest log-likelihood ratio together with the corresponding top four most co-occurring terms. Since these are the terms with the highest abnormal behavior in their current frequency with respect to their historical frequency, we define them as events. Note that in contrast to the original technique that detected events for pre-defined geographical areas, we adjusted the approach to calculate the log-likelihood measure for the frequency of all distinct terms in the current time window against their frequency in the past time windows.

ENB is a reimplementation of Alvanaki *et al.* [2], which uses non-overlapping windows to compute statistics about tags and tag pairs. An event consists of a pair of tags and at least one of the two tags needs to be a so-called seed tag. Seed tags are determined by calculating a popularity score. Tags with a popularity

[1] https://dev.twitter.com (April 28, 2016).

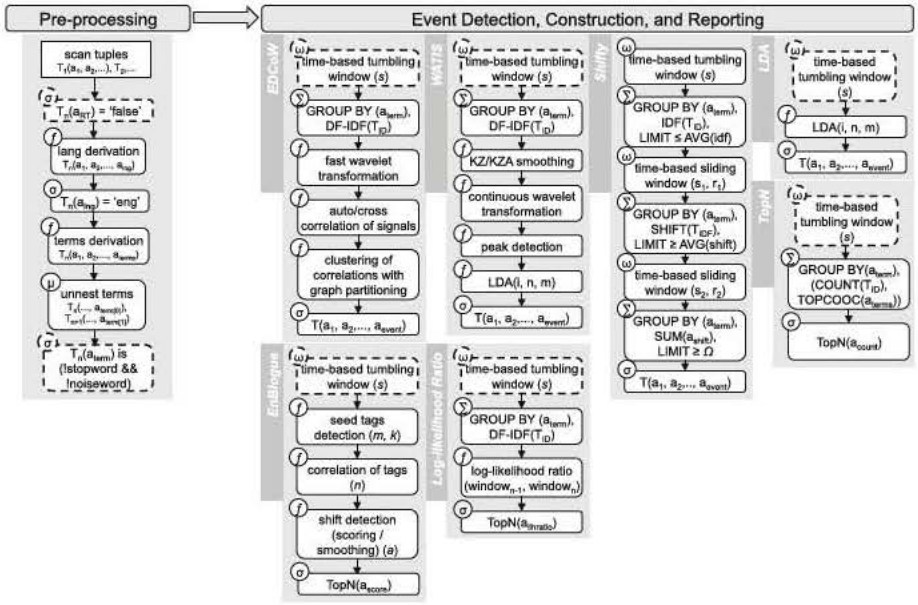

Fig. 1. Query plans of the studied event detection techniques and baselines.

score within a pre-defined range of the top percentage terms (threshold k) are then chosen as seeds. Also a minimum of m tweets need to contain the tag. The correlation of two tags is calculated by a local and global impact factor, which is based on the corresponding sets of tweets that are currently present in the window. If two tags are strongly connected, they are considered to be related. A minimum of n correlations needs to exist. An event is considered as emergent, if its behavior deviates from the expected. In order to detect this condition, the shifting behavior of correlations between terms is calculated by a scoring and smoothing function, which uses the fading parameter a to smooth out past values. Since we require all event detection techniques to output an event as a set of five terms, the three most co-occurring terms of both tags of the pair computed by *ENB* are added to the event. Finally, the technique reports the top N events, which are selected by ranking all events based on the calculated score of shift.

Apart from these event detection techniques, we also implemented two baseline techniques. The *TopN* technique just reports the most frequent N terms per window including their most frequent co-occurrence terms. The *LDA* technique reports topics created by the well-known Latent Dirichlet allocation modeling [8] and is realized by using the Mallet toolkit [17].

3 Results

In this section, we present the results of our experiments as averaged results over all three data sets. First, the impact of changes to the pre-processing phase is studied. Second, we demonstrate the impact of changes to the event detection phase, in particular when varying the size of the time windows.

3.1 Impact of Pre-processing Variations

We evaluate the impact of changes to the pre-processing phase by starting from the parameter settings used in our previous evaluations [26,27]. In the first experiment, we remove the pre-processing operator that suppresses retweets in the input (first operator with a dashed frame in Fig. 1). The results shown in Fig. 2 demonstrate that the inclusion of retweets has a strong impact on the events detected by the studied event detection techniques. In contrast, the influence of this change on the baseline techniques is less pronounced. In the second experiment, we omitted the pre-processing operator that removes stopwords from the input (second operator with a dashed frame in Fig. 1). Figure 3 indicates that the results of the event detection techniques are more stable w.r.t. this second change, with the statistical methods *LLH* and *ENB* proving the most stable. We can also observe that the baseline techniques are more strongly influenced by the inclusion of stopwords than the event detection techniques.

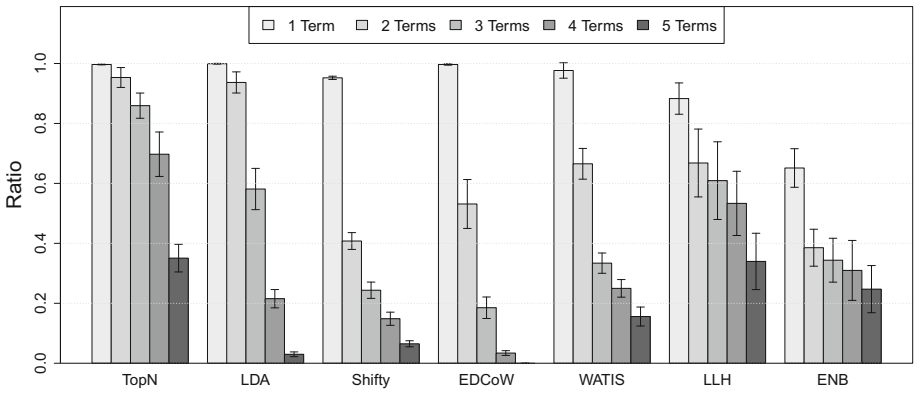

Fig. 2. Impact of including retweets during pre-processing, represented as the ratio of events contained in the results with and without retweets. Each bar presents the ratio of events that share the corresponding number of terms.

Additionally, we measured the throughput (see Fig. 4) for these four different configurations. In the first experiment, the throughput of all techniques decreased by about 30 % to 40 % if retweets are included. In the second experiment, the inclusion of stopwords decreases the throughput by about 10 %, with the exception of *LDA*, where it decreases by almost 30 %.

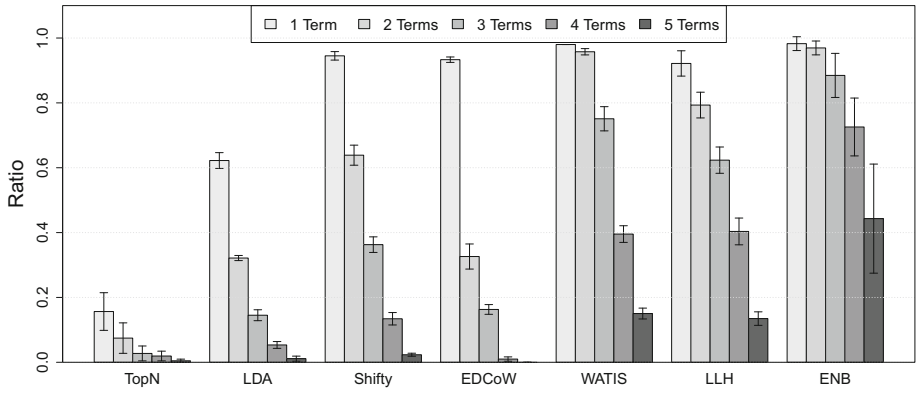

Fig. 3. Impact of including stopwords during pre-processing, represented as the ratio of events contained in the results with and without stopwords. Each bar present the ratio of events that share the corresponding number of terms.

The results observed in these first experiments are as expected. All studied event detection techniques use some form of relative term frequency as a measure for term importance or popularity. In this setting, the inclusion of retweets increases the frequency of terms that are also present if retweets are suppressed. In some cases, this repetition of terms will help to identify an already identified event more clearly. However, since retweets are also heavily used in promotion and advertising, including them can also lead to false positives, *i.e.*, detected events that would be considered "spam". In contrast, the inclusion of stopwords has a different effect as these terms are not present otherwise and therefore do not influence the frequency of event terms. Furthermore, since stopwords are uniformly distributed in the stream, they are unlikely to be identified as an event term themselves. Finally, it is noteworthy that seemingly similar changes to the pre-processing stage can have very different effects.

3.2 Impact of Window Size Variations

We evaluate changes in the event detection phase by varying the window size, which in our previous experiments was set to 1 h. We study the stability of the results by comparing three different configurations with 15, 30, and 60 min, respectively (operators with a dashed frame on the right side of Fig. 1). For techniques that report the top N events as results, we adjust the value of N in accordance to the window size: for 15 min windows the top 5, for 30 min the top 10 and for 60 min the top 20 events are reported. Since the number of events reported per time window can differs substantially depending on the length of the time window, we also adjusted further parameters (*cf.* Table 1). Note that *Shifty* is designed to be independent of the input window size and therefore we have to explicitly stop and restart the processing after 15, 30, or 60 min in order to obtain comparable results.

Table 1. Parameter settings for *Shifty*, *WATIS*, and *EDCoW*.

Technique	Parameters
Shifty15	$s_{input} = 1$ min, $s_1 = 2$ min, $r_1 = 1$ min, $s_2 = 4$ min, $r_2 = 1$ min, $\Omega = 23$
Shifty30	$s_{input} = 1$ min, $s_1 = 2$ min, $r_1 = 1$ min, $s_2 = 4$ min, $r_2 = 1$ min, $\Omega = 22$
Shifty60	$s_{input} = 1$ min, $s_1 = 2$ min, $r_1 = 1$ min, $s_2 = 4$ min, $r_2 = 1$ min, $\Omega = 24$
WATIS15	$s = 25$ s, $N = 3$ intervals, $i_{\text{kza}} = 5$, $i_{\text{lda}} = 500$
WATIS30	$s = 49$ s, $N = 3$ intervals, $i_{\text{kza}} = 5$, $i_{\text{lda}} = 500$
WATIS60	$s = 87$ s, $N = 5$ intervals, $i_{\text{kza}} = 5$, $i_{\text{lda}} = 500$
EDCoW15	$s = 4$ s, $N = 32$ intervals, $\gamma = 2.0$, $\epsilon = 0.1$
EDCoW30	$s = 4$ s, $N = 32$ intervals, $\gamma = 1.5$, $\epsilon = 0.1$
EDCoW60	$s = 4$ s, $N = 32$ intervals, $\gamma = 0.9$, $\epsilon = 0.1$

Figure 5 summarizes the results for this experiment. We can observe that results of the techniques that report a fixed number of N events are more stable than the threshold-based techniques. We can also see that the results of the baseline techniques are very stable in comparison to the results of the event detection techniques. This outcome is explained by the fact that both baseline techniques simply report the most frequent terms, which are bound to be similar in the context of Twitter and independent of a given time frame. Finally, we can observe that *Shifty* is more stable than both *EDCoW* and *WATIS*, which is noteworthy because we introduced artificial interruptions into *Shifty*'s processing to obtain comparable results. By breaking up larger windows into smaller ones, it is possible that *Shifty* misses events that occur across the boundaries of the smaller windows, but would be included entirely in the larger window. Since this effect will increase result instability, *Shifty*'s high stability is a promising result.

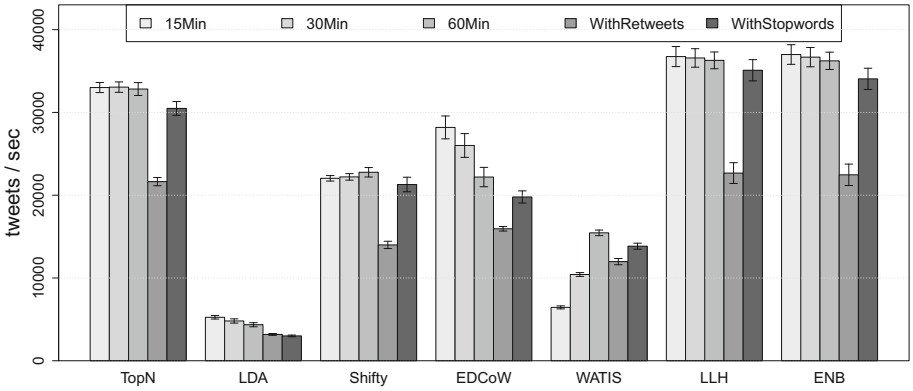

Fig. 4. Impact of all variations for the throughput in tweets/sec.

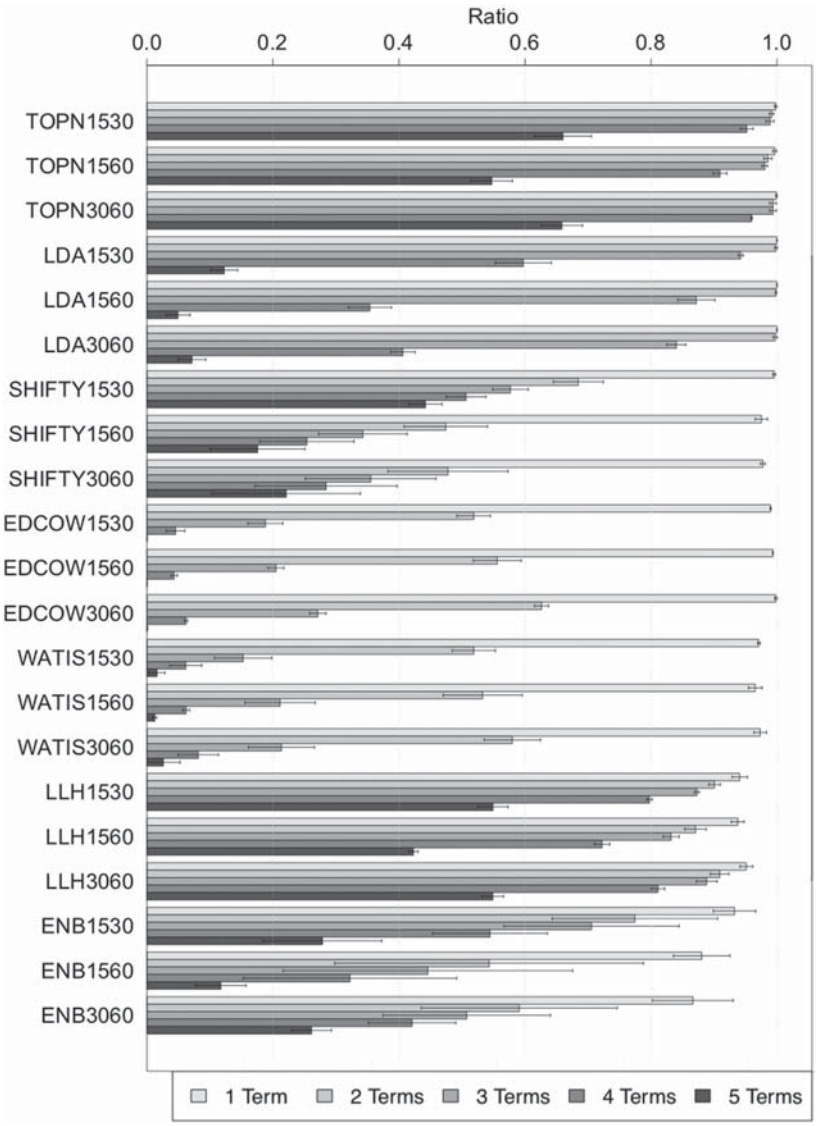

Fig. 5. Impact of different window sizes during event detection, represented as the ratio of events that are contained in all results (*e.g.*, 15 to 30 min, 15 to 60 min, and 30 to 60 min). Each bar present the ratio of events that share the corresponding number of terms.

Again, we also measured the throughput (see Fig. 4) achieved by the different techniques in each configuration. In all our experiments, the throughput of the baselines techniques, as well as the one of *Shifty*, *LLH*, and *ENB* remained stable across the window sizes that we tested. This is due to the fact that the three event detection techniques apply various filtering steps early on and thereby keep the number of terms to analyze within a certain lower bound. The first exception to this observation is *WATIS*. The throughput of *WATIS* when using 30 min windows is twice as high as when using 15 min windows. In the case of 1 h windows, the throughput of *WATIS* is almost three times higher as when using 15 min windows. This is attributable to the processing time of *WATIS* strongly correlating with the number of terms entering the analysis phase, which itself depends on the window size. In the case of 30 min windows, almost twice as many terms are processed as in the case of 15 min. For 1 h windows, the number of terms is three times higher than for 15 min windows. The second exception is *EDCoW*, which exhibits the opposite behavior of *WATIS*, *i.e.*, throughput decreases for longer windows w.r.t. shorter ones. The two most important factors contributing to the run-time of *EDCoW* are the computation of the auto-correlation and the graph partitioning (*cf.* Fig. 1). In the case of the auto-correlation computation, longer windows produce longer signals, which require more time to be processed than shorter signals. The complexity of the graph partitioning also increases with longer windows, since a 15 min window consists on average of about 12,000 edges, while the graphs for 30 min and 1 h windows contain an average of about 25,000 and 70,000 edges, respectively.

4 Conclusions and Future Work

In this paper, we addressed the evaluation of event detection techniques w.r.t. their result stability in an effort to study the reproducibility of experiments in this research area. Our results show that minor modifications in the different phases of the techniques can have a strong impact on the stability of their results. However, we must take into account that by changing the size of the windows, the existing terms in the time frame can vary considerably. Therefore, it is to be expected that the ratio for 3 to 5 terms is very low. Also, the event detection techniques *WATIS* and *EDCoW* are originally designed to analyze even longer time frames, such as days, weeks, and months.

As immediate future work, we plan to take advantage of our platform-based approach to extend our evaluations and study further techniques. As extensions of our evaluations we plan to include further parameter settings and to research the interdependencies of the parameters. By reviewing the surveys of related work (*e.g.*, Nurwidyantoro and Winarko [20], Madani *et al.* [15], or Farzindar and Khreich [10]), we found several candidates for this venture. On the one hand, techniques such as *TwitterMonitor* [16] and *Twevent* [13] are interesting because the techniques they use are closely related to our own techniques. On the other hand, clustering and hashing techniques, such as *ET* [21] or the work of Petrović *et al.* [22] would also be interesting to compare. Since the source

code of most of these works is not provided by their authors, it is a challenging task to correctly implement these techniques. Notable exceptions to this lack of reproducibility are *SocialSensor* [1] and *MABED* [11], which are both freely available as source code. In this context, it would be interesting to define measures, which can be used to rank the degree of reproducibility of existing and future research work in the area of event detection. For this purpose, we created a survey [24] about the techniques and evaluations of 42 related works. With this list we can, for example, rank the works based on the availability of source code, pseudo code, or at least a very precise description of the algorithm. Furthermore, the research work could be ranked according to how many parameters the event detection technique needs and how easily the evaluation can be reproduced.

Acknowledgement. The research presented in this paper is funded in part by the Deutsche Forschungsgemeinschaft (DFG), Grant No. GR 4497/4: "Adaptive and Scalable Event Detection Techniques for Twitter Data Streams" and by a fellowship within the FITweltweit programme of the German Academic Exchange Service (DAAD). We would also like to thank the students Christina Papavasileiou, Harry Schilling, and Wai-Lok Cheung for their contributions to the implementations of *WATIS*, *EDCoW*, and *enBlogue*.

References

1. Aiello, L.M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I.: Sensing trending topics in Twitter. IEEE Trans. Multimedia **15**(6), 1268–1282 (2013)
2. Alvanaki, F., Michel, S., Ramamritham, K., Weikum, G.: See what's enBlogue: real-time emergent topic identification in social media. In: Proceedings of International Conference on Extending Database Technology (EDBT), pp. 336–347 (2012)
3. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: real-world event identification on Twitter. In: Proceedings of International Conference on Weblogs and Social Media (ICWSM), pp. 438–441 (2011)
4. Beel, J., Breitinger, C., Langer, S., Lommatzsch, A., Gipp, B.: Towards reproducibility in recommender-systems research. User Model. User-Adap. Inter. **26**(1), 69–101 (2016)
5. Beel, J., Langer, S.: A Comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems. In: Kapidakis, S., Mazurek, C., Werla, M. (eds.) TPDL 2015. LNCS, vol. 9316, pp. 153–168. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24592-8_12
6. Beel, J., Langer, S., Genzmehr, M., Nürnberger, A.: Introducing Docear's research paper recommender system. In: Proceedings of Joint Conference on Digital Libraries (JCDL), pp. 459–460 (2013)
7. Bethard, S., Jurafsky, D.: Who should i cite: learning literature search models from citation behavior. In: Proceedings of International Conference on Information and Knowledge Management (CIKM), pp. 609–618 (2010)
8. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
9. Cordeiro, M.: Twitter event detection: combining wavelet analysis and topic inference summarization. In: Proceedings of Doctoral Symposium on Informatics Engineering (DSIE) (2012)

10. Farzindar, A., Khreich, W.: A survey of techniques for event detection in Twitter. Comput. Intell. **31**(1), 132–164 (2015)
11. Guille, A., Favre, C.: Mention-anomaly-based event detection and tracking in Twitter. In: Proceedings of International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 375–382 (2014)
12. He, Q., Pei, J., Kifer, D., Mitra, P., Giles, L.: Context-aware citation recommendation. In: Proceedings of International Conference on World Wide Web (WWW), pp. 421–430 (2010)
13. Li, C., Sun, A., Datta, A.: Twevent: segment-based event detection from tweets. In: Proceedings of International Conference on Information and Knowledge Management (CIKM), pp. 155–164 (2012)
14. Lu, Y., He, J., Shan, D., Yan, H.: Recommending citations with translation model. In: Proceedings of International Conference on Information and Knowledge Management (CIKM), pp. 2017–2020 (2011)
15. Madani, A., Boussaid, O., Zegour, D.E.: What's happening: a survey of tweets event detection. In: Proceedings of International Conference on Communications, Computation, Networks and Technologies (INNOV), pp. 16–22 (2014)
16. Mathioudakis, M., Koudas, N.: TwitterMonitor: trend detection over the Twitter stream. In: Proceedings of International Conference on Management of Data (SIGMOD), pp. 1155–1158 (2010)
17. McCallum, A.K.: MALLET: A Machine Learning for Language Toolkit (2002). http://mallet.cs.umass.edu
18. McCreadie, R., Soboroff, I., Lin, J., Macdonald, C., Ounis, I., McCullough, D.: On building a reusable Twitter corpus. In: Proceedings of International Conference on Research and Development in Information Retrieval (SIGIR), pp. 1113–1114 (2012)
19. McMinn, A.J., Moshfeghi, Y., Jose, J.M.: Building a large-scale corpus for evaluating event detection on Twitter. In: Proceedings of International Conference on Information and Knowledge Management (CIKM), pp. 409–418 (2013)
20. Nurwidyantoro, A., Winarko, E.: Event detection in social media: a survey. In: Proceedings of International Conference on ICT for Smart Society (ICISS), pp. 1–5 (2013)
21. Parikh, R., Karlapalem, K.: ET: Events from Tweets. In: Proceedings of International Conference Companion on World Wide Web (WWW), pp. 613–620 (2013)
22. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to Twitter. In: Proceedings of Conference on the North American Chapter of the Association for Computational Linguistics (HLT), pp. 181–189 (2010)
23. Petrović, S., Osborne, M., Lavrenko, V.: Using paraphrases for improving first story detection in news and Twitter. In: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pp. 338–346 (2012)
24. Weiler, A.: Design and evaluation of event detection techniques for social media data streams. Ph.D. thesis, University of Konstanz, Konstanz (2016)
25. Weiler, A., Grossniklaus, M., Scholl, M.H.: Event identification and tracking in social media streaming data. In: Proceedings of EDBT Workshop on Multimodal Social Data Management (MSDM), pp. 282–287 (2014)
26. Weiler, A., Grossniklaus, M., Scholl, M.H.: Evaluation measures for event detection techniques on Twitter data streams. In: Maneth, S. (ed.) BICOD 2015. LNCS, vol. 9147, pp. 108–119. Springer, Heidelberg (2015). doi:10.1007/978-3-319-20424-6_11

27. Weiler, A., Grossniklaus, M., Scholl, M.H.: Run-time and task-based performance of event detection techniques for Twitter. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 35–49. Springer, Heidelberg (2015). doi:10.1007/978-3-319-19069-3_3
28. Weiler, A., Scholl, M.H., Wanner, F., Rohrdantz, C.: Event identification for local areas using social media streaming data. In: Proceedings of SIGMOD Workshop on Databases and Social Networks (DBSocial), pp. 1–6 (2013)
29. Weng, J., Lee, B.S.: Event detection in Twitter. In: Proceedings of International Conference on Weblogs and Social Media (ICWSM), pp. 401–408 (2011)
30. Zarrinkalam, F., Kahani, M.: SemCiR - a citation recommendation system based on a novel semantic distance measure. Program: Electron. Libr. Inf. Syst. **47**(1), 92–112 (2013)