Information Systems ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

An evaluation of the run-time and task-based performance of event detection techniques for Twitter

Andreas Weiler *, Michael Grossniklaus, Marc H. Scholl

Department of Computer and Information Science, University of Konstanz, Germany

ARTICLE INFO

Keywords:
Event detection
Performance evaluation
Twitter social media data stream

ABSTRACT

Twitter's increasing popularity as a source of up-to-date news and information about current events has spawned a body of research on event detection techniques for social media data streams. Although all proposed approaches provide some evidence as to the quality of the detected events, none relate this task-based performance to their run-time performance in terms of processing speed, data throughput, or memory usage. In particular, neither a quantitative nor a comparative evaluation of these aspects has been performed to date. In this article, we study the run-time and task-based performance of several state-of-the-art event detection techniques for Twitter. In order to reproducibly compare run-time performance, our approach is based on a general-purpose data stream management system, whereas task-based performance is automatically assessed based on a series of novel measures.

1. Introduction

With 271 million monthly active users[1] that produce over 500 million tweets per day,[2] Twitter is the most popular and fastest-growing microblogging service. Microblogging is a form of social media that enables users to broadcast short messages, links, and audiovisual content. In the case of Twitter, these so-called *tweets* can contain 140 characters[3] and are posted to a network of *followers* as well as to a user's public timeline. The brevity of tweets makes them an ideal mobile communication medium and Twitter is therefore increasingly used as an information source for current events as they unfold. For example, Twitter data has been used to detect earthquakes [41], to track epidemics [16], or to monitor elections [51].

In this context, an *event* is defined as a real-world occurrence that takes place in a certain geographical location and over a certain time period [5]. For traditional media such as newspaper archives and news websites, the problem of event detection has been addressed by research from the area of Topic Detection and Tracking (*TDT*). However, topic detection in Twitter data streams introduces new challenges. First, Twitter "documents" are much shorter than traditional news articles and therefore harder to classify. Second, tweets are not redacted and thus contain a substantial amount of spam, typos, slang, etc. Finally, the rate at which tweets are produced is very bursty and continually increases as more people adopt Twitter every day.

Several techniques for event detection in Twitter have been proposed. However, most of these approaches suffer from two major shortcomings. First, they tend to focus exclusively on the information extraction aspect and

* Corresponding author.
E-mail addresses: Andreas.Weiler@uni-konstanz.de (A. Weiler), Michael.Grossniklaus@uni-konstanz.de (M. Grossniklaus), Marc.Scholl@uni-konstanz.de (M.H. Scholl).

[1] http://www.statista.com/study/9920/twitter-statista-dossier/ (August 18, 2015).

[2] http://www.sec.gov/Archives/edgar/data/1418091/000119312513390321/d564001ds1.htm (August 18, 2015).

[3] In August 2015, Twitter lifted this restriction for direct messages, but not for general tweets: http://blog.twitter.com/2015/removing-the-140-character-limit-from-direct-messages (August 18, 2015).

http://dx.doi.org/10.1016/j.is.2016.01.003
0306-4379/

often ignore the streaming nature of the input. As a consequence, they make unrealistic assumptions, which limit their practical value. Examples of such assumptions include buffering entire months of Twitter data before processing it or fixing a complex set of parameters at design-time using sample data. Second, very few authors have evaluated their technique quantitatively or comparatively. While most provide some qualitative evidence demonstrating their task-based performance, very few consider run-time performance. Therefore, little or no research to date has measured the computing cost of the same result quality for different approaches. We argue that understanding this trade-off is particularly important in a streaming setting, where processing needs to happen in real-time.

In this article, we present a method to evaluate the run-time and the task-based performance of current and future event detection techniques. In order to measure comparable run-time performance numbers, we propose to "standardize" event detection techniques by implementing them based on a single data stream management system. Additionally, we developed several scalable measures to assess the task-based performance of event detection techniques automatically, i.e., without painstakingly crafting a gold standard manually. The specific contributions of this article are as follows.

1. Streaming implementations of state-of-the-art event detection techniques for Twitter that are consistent with respect to each other.
2. Detailed study of the task-based and run-time performance of well-known event detection techniques.
3. Platform-based approach that will enable further systematic performance studies for novel event detection techniques in the future.

This article is an extended presentation of Weiler et al. [49]. In comparison to the conference version, it features two additional contributions. First, this article provides a much more detailed survey of the state of the art in event detection techniques for Twitter data streams. Since we analyze several recent approaches that have not yet been discussed in other surveys, this article closes a gap to these surveys with respect to open-domain event detection techniques. Second, this article studies the run-time and task-based performance of event detection techniques more in-depth by applying extra measures. In terms of run-time performance, we additionally examine the run-time performance on a second hardware setting and also the memory requirements of each technique, whereas in terms of task-based performance, we analyze how many repeated as well as common events are detected by the techniques.

The remainder of this article is structured as follows. Section 2 presents our survey of the state of the art in open-domain event detection for Twitter data streams. In Section 3, we give a brief overview of Niagarino, the data stream management system that we used as an implementation platform. Section 4 describes the selected event detection techniques and their streaming implementations using Niagarino. Section 5 discusses the results of the evaluation that we performed in order to study the selected task-based and run-time performance of these event detection techniques. Finally, concluding remarks are given in Section 6.

2. Background

In recent years, a lot of research has been conducted in the area of event detection and tracking techniques for Twitter. Consequently, a number of surveys exist that document the current state of the art. For example, Survey 1 by Nurwidyantoro and Winarko [36] summarizes eleven techniques to detect disaster, traffic, outbreak, and news events. Survey 2 by Madani et al. [28] presents 13 techniques that each address one of the four challenges of health epidemics identification, natural events detection, trending topics detection, and sentiment analysis. A more general survey with a wide variety of research topics related to sense making in social media data is Survey 3 by Bontcheva and Rout [12]. The work defines five key research questions—user, network, and behavior modeling as well as intelligent and semantic-based information access. The part about semantic-based information access also includes an overview about event detection techniques in social media data streams. They classify event detection methods into three categories: clustering-based, model-based, and those based on signal processing. Furthermore, an overview about techniques for "sub-events" detection is presented. Finally, the most extensive survey to date is Survey 4 by Farzindar and Khreich [17] with a listing of 16 different techniques categorized by their detection methods, tasks, event types, application domains, and evaluation metrics.

Since the work presented in this paper targets approaches that support the detection of general (unknown) events [5], we will focus our survey on open-domain event detection techniques that share this goal. To the best of our knowledge, Table 1 lists all existing approaches that fall into this category in ascending order of their year of publication. The table also summarizes what technique approaches to use to detect events. Finally, the last column indicates in which of the above-mentioned surveys each approach is included. The four surveys are referred to by using the number assigned to them in the previous paragraph.

As shown in Table 1, many approaches use similar techniques, but with individual modifications or extensions. For example, several approaches are based on statistical models that are defined to detect bursty behavior of single terms or pairs of terms. Most of these approaches analyze the Document Frequency (DF) or the Inverse Document Frequency (IDF) [43] of terms over time. While this technique is very common, the precise definition of what is considered to be a "burst" varies significantly among approaches. Another degree of variation is the level of sophistication present in existing approaches, which ranges from wavelet analysis to simple threshold-based decisions. Some approaches additionally include spatial information in the analysis and present event detection techniques that are specific to a local area. Finally, several approaches can be summarized as using term clustering