Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 117 (2017) 248–255

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Arabic Computational Linguistics, ACLing 2017, 5-6 November 2017, Dubai, United Arab Emirates

# Towards Efficient Online Topic Detection through Automated Bursty Feature Detection from Arabic Twitter Streams

Muhammad Hammad[a] and Samhaa R. El-Beltagy[b] [1]

[a]Cairo University, Giza 12613, Egypt
[b] Center for Informatics Science, Nile University, Giza 12613, Egypt

## Abstract

Detecting trending topics or events from Twitter is an active research area. The first step in detecting such topics focuses on efficiently capturing textual features that exhibit an unusual high rate of appearance during a specific timeframe. Previous work in this area has resulted in coining the term "detecting bursty features" to refer to this step. In this paper, TFIDF, entropy, and stream chunking are adapted to investigate a new technique for detecting bursty features from an Arabic Twitter stream. Experimental results comparing bursty features extracted from Twitter streams, to Twitter's trending Hashtags and headlines from local news agencies during the same time frame from which tweets were collected, show a great deal of overlap indicating that the presented algorithm is capable of detecting meaningful bursty features.

Peer-review under responsibility of the scientific committee of the 3rd International Conference on Arabic Computational Linguistics.

*Keywords*: topic detection; topic tracking; bursty; twitter

## 1. Introduction

With the rise of Web 2.0, social networks and micro-blogging services have boomed. These mediums have become incredibly popular and have been able to create a huge user base that is continuously growing and generating a wealth of online content. According to Statista[1], there were 1.5B and 320M active users on Facebook and Twitter respectively as of Jan 2016. These figures encompass users across the globe, posting and reading in various languages. The focus of this work however, is on the Arabic language only.

A study published by Semiocast [2] in 2012, reported that Arabic was the fastest growing language on Twitter during that year. Another study conducted in 2015 by the Mohammed Bin Rashid School of Government [3] put the number of active Facebook users at 18.3 million and the number of active Twitter users at 5.8 million at the end of May 2014,.

Twitter enables users to post status updates, or tweets, no longer than 140 characters to a network of followers using various communication services (e.g., cell phones, emails, Web interfaces, or other third-party applications). Some users view the 140-character limit as an inconvenient constraint from Twitter. However, many other users see that short information as easier to understand and faster to spread. Because of the nature of Twitter, it is usually the first place real time events are reported. As a result, much research has been carried out over the past few years to investigate how to quickly detect certain critical events such as earthquakes [4] from Twitter, with more recent research focusing on the general area of real time event detection from Twitter.

Central to any event detection task, is the sub-task of identifying keywords that suddenly start appearing at a much higher rate than their average. The faster a system can identify those, the more effective it could be for real time event detection. The goal of this work is to identify these keywords, which are also known as bursty keywords or features, from a stream of Arabic tweets. The proposed approach uses statistical analysis to discover features that suddenly exhibit a surge.

The rest of this paper is organized as follows; section 2 overviews related work, section 3 describes the proposed approach, section 4 describes the dataset used for experimentation and evaluation, section 5 presents the evaluation results, and finally section 6 concludes this paper, and presents future research directions.

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 3rd International Conference on Arabic Computational Linguistics.
10.1016/j.procs.2017.10.116

## 2. Related Work

Many of the ideas presented in this work borrow from the area of Topic Detection and Tracking (TDT). The focus of TDT is on tracking and detecting events or topics in news streams. Most of the approaches introduced for TDT seek to discover terms and features that exhibit an unusually high rate of appearance (bursty features) whether in a stream, or a text corpus for online or offline topic detection. In the various works presented in [11–15], the authors always focused on finding bursty features as a first step. Once those were found, each work presented its own technique for detecting new topics or events in a stream. Most of the research carried out in this area, was on English text.

An example of this work is that by Kleinberg [5] who constructed a 2-state finite automaton model to learn different patterns associated with the sudden appearance of features in a stream. Building on Kleinberg's idea, Kumar et al. [6] applied Kleinberg's algorithm to discover bursty communities in a Weblog graph. Qi He et al [7] also adapted Kleinberg's algorithm to identify bursty features from text streams and used them for document representation instead of the classical vector space model. Wayne Zhao et al. [8] developed BurstVSM; a system for bursty feature representation. In their work, the authors modified Kleinberg's algorithm to enable it to detect and represent features in the form of triples where each triple consists of a term, a start timestamp and an end timestamp.

Sungjun Lee et al. [9] developed a new technique for bursty event detection. The focus of their work was on disaster management. In this work, a term is considered as a bursty feature based on a score calculated from the multiplication of four values defined by the authors: skewness, consistency, periodicity and variation. Gabriel Pui et al. [10] worked on splitting a stream into non-overlapping time windows of the same size. They then used a generative probabilistic model to compute the probability that a set of documents that contain a given feature exists in a time window. Features are then clustered into events using a clustering algorithm that they have devised and which is called HB-Event.

H. Abdelhaq et al developed EvenTweet [11] which focuses on localized events and describes each event by a tuple consisting of the set of event related keywords, the event's geographic location, and the event's start time. Given a set of keywords that are derived from a set of tweets published in a specific timeframe, they adapted the discrepancy paradigm [12] to capture bursty keywords. This paradigm measures the deviation between each word's usage in the current timeframe and its expected usage within a baseline. The baseline usage is constant and is calculated from the history usage of the word in the previous timeframes. The higher the deviation, the higher the burstiness degree.

C. Li et al developed Twevent [13] which adopted 'tweet segments' as a representation form for tweets instead of unigrams. A tweet segment is defined as one or more consecutive words (or phrase) in a tweet message. Twevent starts by dividing each tweet into a sequence of consecutive and non-overlapping phrases (i.e., segments). Based on the segments' frequency and user frequency (user support; the number of users posting tweets containing the segments), bursty segments are calculated. Bursty segments are then clustered based on the similarity between each pair of segments, and the resulting clusters are then checked against a Wikipedia knowledge base to decide the important events from the trivial ones.

J. Weng et al [14] approached event detection using an approach that carries out clustering of wavelet-based Signals (EDCoW). The approach creates signals for individual words over time, by monitoring their frequency over a specific time interval. The similarity between signals (words) i computed using cross correlation between each two signals. Signals are then clustered based on similar bursty patterns. TopicSketch [15], is a model that adopts a novel data sketch method. In this method, data is captured from a Twitter stream and then modeled as a matrix of words represented by their frequencies. The matrix is updated every time a new tweet arrives. Using the Inhomogeneous Poisson Process [16], the presented approach infers (K) active topics, where (K) is a predefined upper bound value for the number of topics that can occur at time stamp (t). The inference is carried out as an optimization problem for a set of equations.

Daehoon Kim et al. [17] performed simple syntactic feature–based filtering to select candidate keywords from a Twitter stream. Then, using several heuristics the authors merge the keywords, and select the top bursty features based on their term frequency. Jheser G. and Barbara P. [18] proposed a methodology based on time-window analysis. In their approach, they combine the keywords from each window in a single bag of words, insert them in a Hash Table, and finally find the bursty keywords by removing from the hash table the keywords that don't have a positive relevance variation based on their own criteria.

## 3. Proposed Work

As stated in the introduction, the aim of this work is to identify bursty keywords from an Arabic twitter stream. The proposed approach uses statistical analysis to discover features that suddenly exhibit a surge. In the context of this work, a bursty feature may be a single word, a multi-token term, a hashtag, or a phrase. The main idea of the work is to extract features from tweets that have been streamed within a short time frame defined by a variable **W** and to compare those to features collected from other features collected over a much larger time frame **F.** TFIDF and entropy are then used to derive bursty features based on the features extracted from both **W** and **F**.

An overview of the system is shown in figure 1

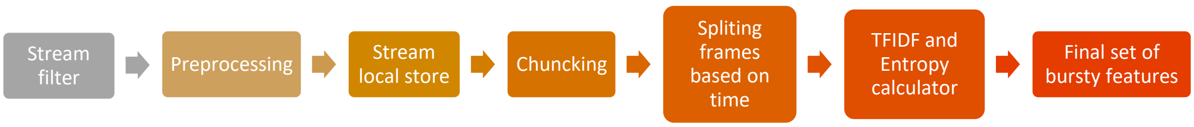
Stream filter
Preprocessing
Stream local store
Chuncking
Spliting frames based on time
TFIDF and Entropy calculator
Final set of bursty features

*Figure 1 - System Overview*