Twitter event detection: combining wavelet analysis and topic inference summarization

Mário Cordeiro

Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL
pro11001@fe.up.pt

Abstract. Today streaming text mining plays an important role within real-time social media mining. Given the amount and cadence of the data generated by those platforms, classical text mining techniques are not suitable to deal with such new mining challenges. Event detection is no exception, available algorithms rely on text mining techniques applied to pre-known datasets processed with no restrictions about computational complexity and required execution time per document analysis.
This work presents a lightweight event detection using wavelet signal analysis of hashtag occurrences in the twitter public stream. It also proposes a strategy to describe detected events using a Latent Dirichlet Allocation topic inference model based on Gibbs Sampling. Peak detection using Continuous Wavelet Transformation achieved good results in the identification of abrupt increases on the mentions of specific hashtags. The combination of this method with the extraction of topics from tweets with hashtag mentions proved to be a viable option to summarize detected twitter events in streaming environments.

Keywords: event detection, topic detection, text stream mining, twitter

1 Introduction

Twitter differs from other social networks by being a micro-blogging service that limits the size of messages. This feature that allow twitter users to publish short messages, in a faster and summarized way, make it the preferred tool for the quick dissemination of information over the web. In March 2011, the estimated number of twitter users was 200 million [22] and the amount of messages published in a single day totalized 177 million tweets sent on the March 11, 2011 [23]. People use twitter to share advice, opinions, news, moods, concerns, facts, rumors, and everything else imaginable. Corporations use twitter to make announcements of products, services, events, and news media companies use twitter to publish near real-time information about breaking news.

From the point of view of data mining, tweets can be seen as a source of data enabling users and corporations to stay informed of what is happening now or what's being said about them and their brands. Sentiment Analysis and Opinion Mining performed subsets of tweets mentioning the person or product keywords [14] are common text mining problems applied to twitter corpus.

Being Twitter the "what's-happening-right-now" tool [21] and given the nature of it's data – an real-time flow of text messages (tweets) coming from very different sources covering varied kinds of subjects in distinct languages and locations – makes the twitter public stream an interesting data set for event detection based on text mining techniques.

In fact the use and extension of text retrieval and clustering techniques for event detection has long been a research topic [33]. Examples of specific applications to the twitter stream are the first story detection proposed by Petrovic et al. [16] that tries to detect whether users discuss any new event that have never appeared before in Twitter, Weng et al. [27] proposed the detection of generic events using signal analysis on the Singapore General Election 2011, and Sakaki et al. [20] exploit tweets to detect critical events like earthquake .

In twitter event detection, the underlying assumption that some related words would show an increase in the usage when an event is happening is not a viable method [12] . In comparison to traditional event detection from news wire, the twitter stream include a much higher volume of data flooded by high amounts of meaningless messages. According to a study by Pear Analytics [19], about 40% of all the tweets are pointless "babbles" and 37% conversational. Such tweets, that some authors call noise [16], are important to build a user's social presence [11] and may help to understand the impact an event had or how people reacted to it, but normally they affect negatively the performance of event detection algorithms.

2 Related work

Early first story detection systems where based on the representation of documents as vectors in a term space using term frequencies [1, 32]. Applying a distance measure, new documents are compared to their nearest neighbors and if it's distance exceeds an predefined maximum value the document is considered to be a first story. This method implies to have all the document term frequencies in memory, moreover, finding the nearest neighbor for new documents even for optimized solutions don't provide much improvement over a simple linear search [10]. To overcome this, Petrovic et al. [16] proposed an modified locality sensitive hashing (LSH) [7] used as a nearest neighbor search optimization that fulfills the data stream mining requirements by using constant size buckets.

Authors Chen and Roy [5] and Weng et al. [27] assume that the occurrence of an event may be detected by observing abrupt increases on the use keywords related with the event. Weng et al. [27] proposed an wavelet-based twitter event detection. Initially, based on the number of word occurrences over the time, individual signals for each of the words are constructed. Signals are then filtered per wavelet analysis to reveal bursts in the word's appearance and therefore compute the cross-correlation between signals. Finally events are detected by applying a modularity-based graph partitioning clustering algorithm to the signals. Using wavelet analysis it is compatible with the stream mining requirements, but the cross-correlation between signals or the modularity-based graph partitioning

may not meet those requirements in particular the use of limited resources and the capability of working in real-time.

Using topic distributions rather than bags of words to represent documents reduces the lexical variability and retain the overall semantic structure of the corpus [34]. Topic models discover the abstract "topics" that occur in a collection of documents and are able to identify low sets of representative characteristics of the documents in very high dimensional data. Inference for topic models remain computationally expensive even with the recently advances in fast inference latent Dirichlet allocation (LDA) algorithms Blei et al. [3]. SparseLDA [34], FastLDA [17] and O-LDA [4] are sampling-based inference LDA methods using Gibbs sampling that propose efficient topic inference to text streaming collections. Moving from document topics to event detection requires to introduce other attributes such as spatial and temporal parameters to the LDA model. This lead Pan and Mitra [15] to combine the LDA model with temporal segmentation and spatial clustering in two distinct methods: the proposed 3S-LDA is a LDA done in three step, the document topic assignment, a temporal segmentation and finally spacial clustering; the Space-Time LDA is a spatial latent Dirichlet (SLDA) allocation [26] adapted from the detection of segments in images to the detection events in text corpus. Those algorithms present good event detection in the TDT3 [9] and Reuters news dataset but they were not tested in text streaming.

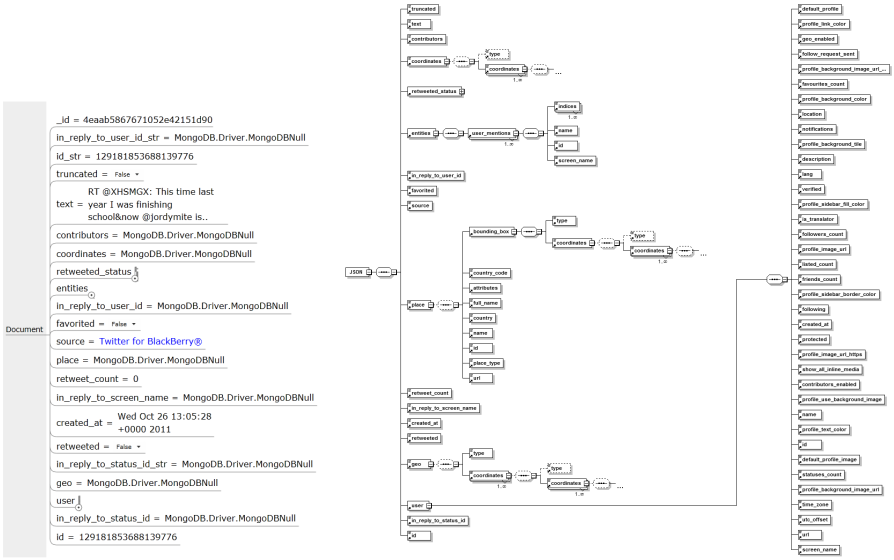
3 The twitter stream

The data set object of analysis will be retrieved using the twitter streaming API (`statuses/sample` method) that, using the default access level ('Spritzer'), returns a random sample of all public tweets [24]. This access level provides a small proportion of all public tweets (1%) [25]. The twitter stream API gives also access the firehose: 100% and gardenhose: 10% streams [24] using special accounts.

The data returned is a set of documents, one per tweet, in the JavaScript Object Notation [6]. These documents (Figure 1a), apart from the text of the tweet, contain additional data like tweet information i.e.: date, source of tweet, type; user information i.e.: profile, location and counters for favourites, friends, followers, etc.; entities mentioned in the tweet text i.e.: urls, hashtags and user, among other information.

Given the average number of 140 million tweets sent per day referred by Twitter [23], it is expected that the size of the data retrieved by the Streaming API (1%), in a 24 hour time span, will be roughly 1.400.000 tweets. The 140-character limit of tweets give an expected 196 MBytes per day or 2269 Bytes per second data stream.

The json tweet document contains attributes describing the tweet, user information, tweet relations with other tweets, a lists of urls, hashtags and user mentions contained in the tweet. In some cases information related with the lo-



(a) tweet json sample (b) tweet structure

Fig. 1: Tweet document structure and json sample

cation of the user is also provided in the document. The Figure 1b shows the list of attributes and the relations between entities in the document.

4 Data stream event detection

Given the volume of data and the real-time nature of the twitter stream, event detection should be processed in an online manner. It's objective is to identify abrupt rises in the use of sets of words that could point to the occurrence of events. Unlike Sentiment Analysis and Opinion Mining algorithms, analysis is not based on a restricted set of tweets mentioning predefined keywords, in this case events are not known *a priori*, so data analysis cannot be confined to filtering techniques using only tweets mentioning sets of keywords. The event detection should address the maximum number of tweets and therefore instead of using classical text mining algorithms it should consider mining algorithms adapted to the processing of data streams.

In conventional machine learning algorithms the training data is available as a whole set. On the opposite, when the training set is a potentially endless flow of data arriving in an order that cannot be controlled, we consider that we are in the presence of a data mining algorithm that is able to learn from a stream. Data stream mining algorithms can be addressed by classical data mining algorithms that met the following requirements [2]:

– **Process an example at time, and inspect it only once:** No random access to the data being supplied. Examples are accepted as they arrive and in the arriving order. Algorithms can remember previous examples but should keep the used resources at a minimum level. Algorithms that require more than one pass to operate are topically not suitable for data streaming mining;
– **Use a limited amount of memory:** Memory usage can be divided in two parts the memory used to store running statistics and the memory used to store the current model;
– **Work in a limited amount of time:** To be capable of working in real-time, it must process the examples as fast or fastest than they arrive;
– **Be ready to predict at any point:** It should be able to produce the best model after seeing any number of examples. The process of generation of the model should be as efficient as possible. Final model generation should be direct and avoiding the re-computation of the model based on running statistics.

5 Twitter event detection based on wavelet analysis of hashtag mentions

None of the event detection methods described in section 2 met all the requirements for data stream mining outlined in section 4, therefore they are not suitable to perform real-time event detection in the twitter stream. Hashtag mentions of each tweet are provided in the tweet document returned by the public stream in node `entities hashtags` list (Figure 1b) and replace word occurrences in tweet text. Building occurrence signals from that hashtags it is more cost effective when compared to the building of tweet text word co-occurrences that require word tokenization and pre-preprocessing of the tweets text to remove irrelevant words and other entities like user mentions, urls and hashtags. Given the amount of data needed to be processed, topic detection applied in real-time to the Spritzer twitter stream [25] may not be feasible even with the SparseLDA [34] or O-LDA [4] that are described as real-time approaches to detect latent topics data streams. The proposed approach relies in the monitoring of hashtag mention signals using wavelet signal analysis. Because topic inference models are computationally expensive and require to be retrained each time a new tweet arrives, they will be only used in a later stage of the processing stage. In the current proposal LDA is applied only to hashtags signals that were identified as events at a given time interval and are used to estimate topics that help to describe the event itself.

Figure 2 shows the proposed workflow to perform the detection of events in the twitter stream. The following sections describe each one of the steps in detail.

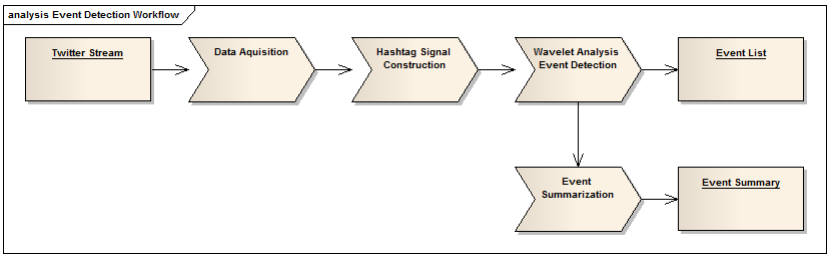


Fig. 2: Event Detection Workflow

5.1 Data acquisition

The data set acquisition was done getting the tweet documents in json format and inserting them in a document-oriented database. Two tools were used in this task, cURL [28] to access the twitter stream and MongoDB [29] to store the data set in a way that can be queried. The Code 1.1 shows the command to insert live twitter data in the database. All the tweets retrieved by the cURL were stored in a collection named `tweets` in a database called `twitter`.

```
curl -k https://stream.twitter.com/1/statuses/sample.json -uUSER:PASS |
    mongoimport -d twitter -c tweets
```

Code 1.1: Data acquisition of twitter data set

5.2 Hashtag signal construction

The event detection of the twitter stream was made by analyzing the evolution of hashtag mentions over the time in the twitter's public stream. There are several transformations necessary to be performed prior to the event detection using wavelet analysis. Event detection is based on the peak analysis of individual hashtags mentions over the time (hashtag signal now on). To build such signals, for each one of the hashtags mentioned in tweets, it was necessary to build individual mention counts for each one of the hashtags over the time.

The data preparation process was made using map reduce transformations [8]. MongoDB supports map reduce transformations using javascript functions [13]. Therefore, all data preparation algorithms were performed on the database side. Processing the twitter stream data source using this approach removed unnecessary data transfers involving high amounts of data transfers across external components and lead to an increase in the performance compared to other approaches. To build the individual hashtags signals it was necessary to perform two map reduce transformations:

Extraction of hashtags from tweet documents The first step was the extraction of the hashtags mentioned in the the tweet documents. Using a map

reduce transformation all hashtags were retrieved from tweets and then grouped in intervals of 5 minutes. The Code 1.2 shows the json document resulted after that transformation for a single 5 minute interval. The json document contains all the hashtags mentioned in the tweets posted between 21:45:00 and 16:50:00 in the 12th November 2011 (Code 1.2 line 1). The `hashtags` list show each one of the hashtags mentioned in the time period (Code 1.2 lines 4 to 9). In the `totaltext` list is shown the tweet text where the hashtag was mentioned. In order to improve the topic inference process described in section 5.4, all hashtags, URLs and user mentions were removed from the tweets text (Code 1.2 lines 11 to 16,). The `totalcount` shows the total number of hashtags mentions in the given time period, Code 1.2 line 17 shows that were used 596 hashtag mentions in this 5 minute period.

```
1  { _id: ISODate(2011-11-12T21:45:00Z),
2    value: {
3      hashtags: [
4        'currucucu',
5        'kappennu',
6        'thingslongerthankimsmarriage',
7        'eribelieberbbb12',
8        [...],
9        'stilldontliketheguy' ],
10     totaltext: [
11       'jajja sii lo poco jaja  porfa saludes a de',
12       'roept geesten op en ze wilt niet stoppen',
13       'the time a black person gets to live in a horror film',
14       'gente usa a tag',
15       [...],
16       'dont know how i feel a bout todd halley'  ],
17     totalcount: 596  } }
```

Code 1.2: Map reduce result of hashtag mentions in an interval of 5 minutes

Construction of individual hashtag mention signals The final map reduce transformation is made to build hashtag mention signals over the time. Two simultaneous transformations were done in this map reduce. It is necessary to count the hashtag mentions in each 5 minute interval and grouping them in separated time series one for each hashtag. In Code 1.3 is shown an example were for each hashtag the `timeline` node contains pairs of timestamps (unix epoch [30]) associated with a count of the number of times the hashtag was mentioned over the time. In Code 1.3 line 1 is shown the time series for the "np" hashtag. The `textline` node contains a concatenation of all the tweets text that had the hashtag mentioned in the given time interval (Code 1.3 lines 11 to 16). This text will be used in section 5.4 to summarize the event with related topics once that hashtags itself may not be self explanatory. Line 18 in Code 1.3 shows the total number of times the hashtag "np" was mentioned in the whole time series (all 5 minutes intervals).

```
1  { _id: 'np',
2    value: {
```

```
3     timeline: {
4       1320183900000: 2,
5       1320184200000: 17,
6       1320184500000: 21,
7       1320184800000: 13,
8       [...],
9       1320188400000: 17  },
10    textline: {
11      1320183900000: 'favor wale illest btch',
12      1320184200000: 'i do no days off long walk jill [...]' ,
13      1320184500000: 'twisting stank curreny [...]' ,
14      1320184800000: 'trey songz missing you remix [...]' ,
15      [...],
16      1320188400000: 'marsha ambrosius late nights and [...]'
17    },
18    totalcount: 20648  } }
```

Code 1.3: Map reduce result of the evolution of hashtag "np" over the time ("np" signal)

5.3 Event detection using wavelet hashtag signal analysis

Signals built from Hashtag count mentions were assumed as being the basis for twitter event detection. It was also assumed that they represent the evolution of trends in the twitter stream. The map reduce process described in section 5.2 produced one signal for each of the hashtags mentioned in the defined time interval. Those signals represent time series in the evolution of topic mentions in the tweet stream. It was considered that an abrupt increase on the mention of a hashtag may result in a possible event that is happening at a given time. Wavelet analysis is a well know signal processing method to detect changes and peaks in signals [5]. The continuous wavelet transform (CWT) construct a time-frequency representation of a signal that offers very good time and frequency localization. Two wavelet tools where used to detect events in the twitter stream: the peak analysis was used to detect peaks in the hashtag signal; local maxima detection was used to detect changes in the hashtag signal. Because of the noisy nature of the tweet stream – some hashtag signals have a high variance between consecutive time intervals – signals were preprocessed using Kolmogorov-Zurbenko Adaptive Filters [31] to retrieve the trend of the hashtag signals (Code 1.4 line 3). This step removed some noise from hashtag signals and lead to a better wavelet peak and local maxima detection. Prior to the continuous wavelet transform (CWT) (Code 1.4 line 6), the signal was transformed in a time series (Code 1.4 lines 4 and 5). Extrema locations (in time and in scale) are calculated based on the CWT of the signal (Code 1.4 line 7). Finally, the peak detection in the time series was done and found the local maxima in the each one of the time series via a CWT tree (Code 1.4 line 8). The event detection algorithm was implemented in R (programming language) [18] and results inserted back into the MongoDB database using the resulting json documents.

```
1 foreach (hashtag_signal in twitter_stream)
2 {
3     kzsignal <- kz ( hashtag_signal )
```

```
4       x <- getTimeInterval (hashtag_signal)
5       y <- signalSeries(kzsignal, x))
6       W <- wavCWT(y, wavelet="gaussian2")
7       W.tree <- wavCWTTree(W)[1:100]
8       p <- wavCWTPeaks(W.tree)
9   }
```

Code 1.4: Pseudocode for event detection in R programming

5.4 Event summarization with LDA topic inference

By monitoring the evolution in hashtag mentions, detected events are a list of hashtags that had a peak in mention counts at a given time interval. Some hashtags may be self explanatory of the event itself like "7bilhoesdepessoasnomundo" that revel tweets related with the 7th Billionth Child Born in 31th October 2011 or "papandreou" in tweets related with the greek PM Georgios Papandreou resignation the 9th November 2011. Other hashtags names representing events may not be self explanatory and may need additional information to describe human perceptible event descriptions. To archive this, in parallel when an event is detected, an topic inference algorithm is applied on all the tweets text related with the hashtag in each one of the time series 5 minutes interval. The idea is to extract latent topics using Latent Dirichlet Allocation (LDA) [3] from the tweets text. This method improved the hashtag description summarize the event with a set of topics inferred from tweets belonging to the time interval were the event occurred.

The process of extracting latent topics is done for each hashtag signal obtained from the process described in section 5.2. Each time interval of the `textline` node is considered as document used to train the LDA model. The Code 1.3 (lines 11 to 16) show the input text documents used in the the topic inference process for the "np" hashtag signal. In each one one of the time intervals was created a document-term matrix to be passed to the LDA algorithm. Topic were estimated by the LDA model using Gibbs Sampling, where each document represents the text of all the tweets referring that hashtags in individual intervals of 5 minutes. After the LDA estimation, in each 5 minute interval of the hashtag signal were retrieved 5 topics representative of the hashtag at that given moment. Code 1.5 line 11 shown the 5 extracted topics (t1 to t5) for hashtag "np" between 22:40 and 22:45 of the 1st November of 2011. Note that in this case the topics vary in time depending on the considered time interval (Code 1.5 line 6 show the extracted topics for the previous 5 minute interval and line 15 with topics for the following 5 minutes). In conjunction with the timestamp of the event detection this process will retrieve the 5 topics relevant to the hashtag in using tweets of the timespan where the event occurred.

```
1 [...],
2 { hashtag: 'np',
3   time: 1320186900000,
4   count: 11,
5   text: 'onetimelt this song never gets [...]',
```

```
6   t1: 'nirvana', t2: 'andando', t3: 'aidonia', t4: 'anastacia', t5: 'np'
        },
7 { hashtag: 'np',
8   time: 1320187200000,
9   count: 19,
10  text: 'make me proud special k placebo ok [...]',
11  t1: 'nirvana', t2: 'ambrosius', t3: 'anything', t4: 'afraid', t5: '
        chainz' },
12 { hashtag: 'np,
13  time: 1320187500000,
14  [...],
15  t1: 'np', t2: 'afraid', t3: 'anything', t4: 'ashanti', t5: 'better'
        },[...]
```

Code 1.5: Summarization of hashtag "np" over the time using Latent Dirichlet Allocation (LDA)

6 Experimental results

The the Spritzer twitter stream [25] was monitored between 00:00 of the 10th of November and 23:59 of 18th of November of 2011 totalizing more than 192 hours of data acquisition. In this time period were retrieved 13.651.464 tweets, this gives almost 72.000 tweets per hour and an average of 1.7 million tweets per day. These values were far above that ones expected by estimations made in section 3. In this time interval were mentioned 493.050 distinct hashtags meaning that were constructed 493.050 hashtags signals to be monitored. Table 1 lists the top 20 most mentioned hashtags in the time interval considered.

Table 1: Top 20 popular hashtags and mentions counts

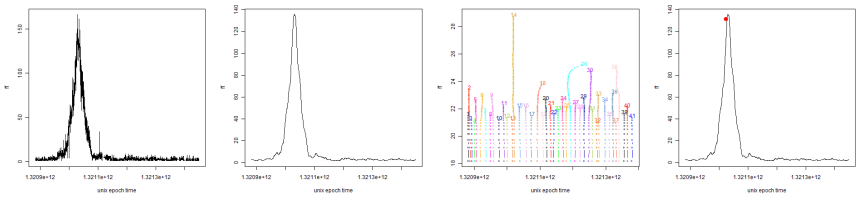
#	hashtag	mentions	#	hashtag	mentions
1	ff	23872	11	xfactor	5929
2	np	20648	12	ows	5800
3	teamfollowback	19088	13	99fm	5511
4	oomf	13687	14	iwannabe	5225
5	useatwitternameinasentence	13640	15	followback	5134
6	nowplaying	11249	16	bahrain	4981
7	nf	7979	17	tfb	4726
8	rt	7741	18	jobs	4632
9	fb	7719	19	myweddingsong	4386
10	thingspeopleshouldntdo	6189	20	peopleschoice	4160

Results of event detection and topic summarization are presented in the following subsections.

6.1 Event detection results

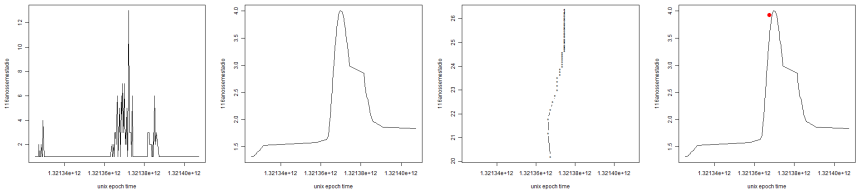
The event detection was done in 4 steps: building of hashtag signals, Kolmogorov-Zurbenko filtering, extrema detection using the continuous wavelet transforma-

tion and peak detection. Figure 3 and Figure 4 show a visual detail of each one of those steps for two hashtags ("ff" and "116anossemestadio"). The red dots shows the detected events at a given timestamp: Figure 3d shows the detection of an event in timestamp `1321029000000` (Friday, 11 Nov 2011 16:30:00 GMT) and Figure 4d an event in timestamp `1321372200000` (Tuesday, 15 Nov 2011 15:50:00 GMT). The "ff" event is related to the FollowFriday twitter trend that occurs every friday and where some twitter users suggest other twitter users to be followed. The "116anossemestadio" is related to the 116th anniversary of the Clube de Regatas do Flamengo Brazilian sports club in November 15, 2011. Remarks that Figure 3c and Figure 4c shown the results of the calculation of all local maxima and local minima for a sliding windows across the signal evolution. Each branch represent a change in the signal evolution. This continuous wavelet transform tree will be the basis used to detect peaks in the signal described in Code 1.4.



(a) original (b) filtered (c) cwt extrema (d) peak detection

Fig. 3: Peak detection for "ff" hashtag



(a) original (b) filtered (c) cwt extrema (d) peak detection

Fig. 4: Peak detection for "116anossemestadio" hashtag

Results of detection of events are presented in Table 2. This table contains a list of hastags and respective event datetime converted from the unix epoch timestamp.

6.2 Event summarization results

Event summarization results are presented in Table 2. The last column of the table shows a summary of the event using 5 topics inferred from tweets that had mentions to the event hashtag in that specific 5 minute interval. Additionally to the hashtag name the infered 5 topics help to summarize the occurrence of the event.

7 Conclusions

The twitter public stream mining using wavelet signal analysis of hashtag occurrences proved to be a valid lightweight option to event detection in this specific real-time and high throughput data stream. The data acquisition of the stream using a document based database, the use of twitter hashtag names included in the stream and the data preprocessing using map reduce transformations performed inside the database, led to an efficient hashtag signal construction. Event detection using peak detection on continuous wavelet transformed hastag signal provided good results in the identification of hashtags mentions bursts independently of being isolated events or events repeated over the time. The proposed technique to describe the detected events using a Latent Dirichlet Allocation topic inference model led to an better description of the event. With this method, detected events by wavelet analysis are enriched with 5 inferred topics from tweets occurred at that given time. Based on the experimental results, in most cases the combination of both hashtag names and inferred topics gave useful description information about the event. For future work and improvements there were identified 3 main topics: hashtag manipulation with the objective to group distinct hashtags related with the same event (“f1jp”, “f1nijigen”, “f1”, “f1chat”, “f1gp”); event repetition learning algorithms to identify and ignore periodic events like the follow friday (“ff”) that result in an event detected every friday; faster topic inference using sampling-based LDA methods applied to text streaming collections like the SparseLDA, FastLDA or O-LDA real-time implementations.

Table 2: List of events detected and respective summarization with topics

hashtag	datetime	5 main topics
116anossemestadio	15 Nov 15:50	flamenguista, 116anossemestadio, estdio, dentro, 116anossemestadio
49ers	13 Nov 23:35	49ers, giants, harbaugh, conversion, touchdown
argentina	11 Nov 21:15	moreno, bolivia, demichelis, jugando, mataaar
berlusconi	12 Nov 20:55	silvio, coglioni, berlusconi, boggles, career
blacksabbath	11 Nov 21:20	bezetting, meninos, blacksabbath, aaaeeewwww, blacksabbath
boosie	14 Nov 17:15	charger, loaded, loaded, better, hunnids
calle13	11 Nov 04:00	actitud, hahahha, celebraron, akabara, cambio
carrierclassic	12 Nov 00:05	correction, basketball, should, uniforms, student
cmaawards	10 Nov 01:40	natasha, watching, watching, taylor, country
cmas	10 Nov 01:50	absolutely, country, country, delicious, aldean
f1	13 Nov 13:05	vettel, actually, forward, campeonato, chequerd
ff	11 Nov 16:30	follow, akexwbc, asegura, ff, excuse
fimdomundoinforma	11 Nov 16:10	sobrevivi, sextafeira, dinheiro, evento, acabar
flamengo116anos	15 Nov 13:05	parabns, estdio, planeta, eterna, parabns
latingrammys	11 Nov 01:20	shakira, franco, premios, bolivar, ascendencia
lilwaynewackestpunchlines	10 Nov 06:45	lilwaynewackestpunchlines, phenomenal, should, phenomenal, almost
mareoflores	13 Nov 22:50	pretenden, siguen, libead, mareoflores, detencin
pacquiao	13 Nov 05:50	marquez, booing, pacquiao, boxing, fighting
raw	15 Nov 01:20	tengok, watching, followers, appearance, boston
tvoh	11 Nov 21:25	degene, kijken, tvoh, ahhaha, geniaal
vergonharecord	14 Nov 00:10	outras, arrependase, tendenciosa, atacando, comprar
veteransday	11 Nov 14:30	country, alabama, served, events, respect
walkingdead	14 Nov 02:45	stupid, walkingdead, alucinaciones, andrea, stupid
wish111111	10 Nov 15:20	tomorrow, person, better, awards, closed
wwe	15 Nov 01:20	cpeatt, welcome, attacked, hometown, wwe
xfactor	13 Nov 17:55	technical, amelia, blowing, devlin, beautiful

References

[1] Allan, J., Lavrenko, V., Malin, D., Swan, R.: Detections, Bounds, and Timelines: UMass and TDT-3 (2007), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.2547

[2] Bifet, A., Kirkby, R.: Data stream mining: a practical approach. Tech. rep., The University of Waikato (Aug 2009)

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

[4] Canini, K.R., Shi, L., Griffiths, T.L.: Online inference of topics with latent dirichlet allocation. Journal of Machine Learning Research - Proceedings Track 5, 65–72 (2009)

[5] Chen, L., Roy, A.: Event detection from flickr data through wavelet-based spatial analysis. In: Cheung, D.W.L., Song, I.Y., Chu, W.W., Hu, X., Lin, J.J. (eds.) CIKM. pp. 523–532. ACM (2009)

[6] Crockford, D.: RFC 4627 - The application/json Media Type for JavaScript Object Notation (JSON). Tech. rep., IETF (2006), http://tools.ietf.org/html/rfc4627

[7] Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: Snoeyink, J., Boissonnat, J.D. (eds.) Symposium on Computational Geometry. pp. 253–262. ACM (2004)

[8] Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Commun. ACM 51(1), 107–113 (2008)

[9] Graff, D., Cieri, C., Strassel, S., Martey, N.: The tdt-3 text and speech corpus. In: in Proceedings of DARPA Broadcast News Workshop. pp. 57–60. Morgan Kaufmann (1999)

[10] Indyk, P., Motwani, R.: Approximate nearest neighbors: Towards removing the curse of dimensionality. In: STOC. pp. 604–613 (1998)

[11] Kaplan, A.M., Haenlein, M.: The early bird catches the news: Nine things you should know about micro-blogging. Business Horizons (Oct 2010), http://dx.doi.org/10.1016/j.bushor.2010.09.004

[12] Kleinberg, J.: Bursty and hierarchical structure in streams. Data Min. Knowl. Discov. 7, 373–397 (October 2003), http://dl.acm.org/citation.cfm?id=861097.861114

[13] MongoDB: Mapreduce — mongodb manual (2011), http://www.mongodb.org/display/DOCS/MapReduce, [Online; accessed 29-December-2011]

[14] Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) LREC. European Language Resources Association (2010)

[15] Pan, C.C., Mitra, P.: Event detection with spatial latent dirichlet allocation. In: Newton, G., Wright, M., Cassel, L.N. (eds.) JCDL. pp. 349–358. ACM (2011)

[16] Petrovic, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: HLT-NAACL. pp. 181–189. The Association for Computational Linguistics (2010)
[17] Porteous, I., Asuncion, A., Newman, D., Smyth, P., Ihler, A., Welling, M.: Fast collapsed gibbs sampling for latent dirichlet allocation. In: In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 569–577 (2008)
[18] R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2010), http://www.R-project.org, ISBN 3-900051-07-0
[19] RyanăKelly: Pearanalytics - twitter study - august 2009 (2009), http://www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf, [Online; accessed 21-November-2011]
[20] Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web. pp. 851–860. WWW '10, ACM, New York, NY, USA (Apr 2010), http://dx.doi.org/10.1145/1772690.1772777
[21] Schonfeld, E.: Techcrunch: Mining the thought stream (2009), http://techcrunch.com/2009/02/15/mining-the-thought-stream, [Online; accessed 14-November-2011]
[22] Shiels, M.: BBC News: Twitter co-founder Jack Dorsey rejoins company. http://www.bbc.co.uk/news/business-12889048 (2011), [Online; accessed 01-November-2011]
[23] Twitter: Twitter Blog: #numbers. http://blog.twitter.com/2011/03/numbers.html (2011), [Online; accessed 01-November-2011]
[24] Twitter: Twitter Developers: Streaming API. https://dev.twitter.com/docs/streaming-api (2011), [Online; accessed 01-November-2011]
[25] Twitter: Twitter Developers: Streaming API Methods. https://dev.twitter.com/docs/streaming-api/methods (2011), [Online; accessed 01-November-2011]
[26] Wang, X., Grimson, E.: Spatial latent dirichlet allocation. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S.T. (eds.) NIPS. MIT Press (2007)
[27] Weng, J., Yao, Y., Leonardi, E., Lee, F.: Event Detection in Twitter. Tech. rep., HP Labs (2011), http://www.hpl.hp.com/techreports/2011/HPL-2011-98.html
[28] Wikipedia: curl — Wikipedia, the free encyclopedia (2011), http://en.wikipedia.org/wiki/CURL, [Online; accessed 01-November-2011]
[29] Wikipedia: Mongodb — Wikipedia, the free encyclopedia (2011), http://en.wikipedia.org/wiki/MongoDB, [Online; accessed 01-November-2011]
[30] Wikipedia: Unix time — Wikipedia, the free encyclopedia (2011), http://en.wikipedia.org/wiki/Unix_time, [Online; accessed 29-December-2011]
[31] Yang, W., Zurbenko, I.: Kolmogorov-zurbenko filters. Wiley Interdisciplinary Reviews: Computational Statistics 2(3), 340–351 (2010), http://dx.doi.org/10.1002/wics.71

[32] Yang, Y., Pierce, T., Carbonell, J.G.: A study of retrospective and on-line event detection. In: SIGIR. pp. 28–36. ACM (1998)
[33] Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 28–36. SIGIR '98, ACM, New York, NY, USA (1998), http://dx.doi.org/10.1145/290941.290953
[34] Yao, L., Mimno, D.M., McCallum, A.: Efficient methods for topic model inference on streaming document collections. In: IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) KDD. pp. 937–946. ACM (2009)