A framework for real-time Twitter data analysis

S. Gaglio, G. Lo Re, M. Morana

In Journal of Computer Communications

Publisher: Elsevier

http://dx.doi.org/10.1016/j.comcom.2015.09.021

A Framework for Real-Time Twitter Data Analysis

Salvatore Gaglio[a,b], Giuseppe Lo Re[a], Marco Morana[a,*]

[a] *DICGIM, University of Palermo, Viale delle delle Scienze, ed. 6 - 90128 Palermo, ITALY*
[b] *ICAR-CNR, Viale delle delle Scienze, ed. 11 - 90128 Palermo, ITALY*

Abstract

Twitter is a popular social network which allows millions of users to share their opinions on what happens all over the world. In this work we present a system for real-time Twitter data analysis in order to follow popular events from the user's perspective. The method we propose extends and improves the Soft Frequent Pattern Mining (SFPM) algorithm by overcoming its limitations in dealing with dynamic, real-time, detection scenarios. In particular, in order to obtain timely results, the stream of tweets is organized in dynamic windows whose size depends both on the volume of tweets and time. Since we aim to highlight the user's point of view, the set of keywords used to query Twitter is progressively refined to include new relevant terms which reflect the emergence of new subtopics or new trends in the main topic. The real-time detection system has been evaluated during the 2014 FIFA World Cup and experimental results show the effectiveness of our solution.

Keywords: Social Sensing, Twitter Analysis, Topic Detection

1. Introduction

Over the last 40 years, providing automatic solutions to analyze text documents collection has been one of the most attractive challenges in the field of information retrieval. More recently, the focus has moved towards dynamic, distributed environments, where documents are continuously created by the users of a virtual community, i.e. the social network. In the case of Twitter, such text portions, called *tweets*, are usually related to events which involve many people in different parts of the world.

In this paper we present a framework to analyze the Twitter stream in order to detect relevant topics within a generic macro event.

Differently from other techniques which focus on the detection of specific events, e.g. earthquakes, or provide offline solutions for the analysis of tweets that match some static filter, our system has been designed to adapt its behavior

[*] Corresponding author

Email addresses: salvatore.gaglio@unipa.it (Salvatore Gaglio), giuseppe.lore@unipa.it (Giuseppe Lo Re), marco.morana@unipa.it (Marco Morana) to the specific nature of incoming data. After choosing some generic terms to query Twitter, the stream of tweets is split into dynamics windows which are analyzed to promptly detect relevant topics. Since our aim is to capture the user's perspective on discussing the events, the initial set of keywords is progressively updated to include new important terms emerging from the tweets, or to delete those unused.

The topic detection algorithm we present is an improved version of Soft Frequent Pattern Mining (SFPM), designed to overcome the limitations of SFPM in dealing with dynamic, real-time, detection scenarios. We chose SFPM as core of our framework for two reasons. First, because of its better performance in detecting relevant topics as compared to other basic techniques [1]; second, because of its modular design, which allows to easily change the way relevant terms are selected, while maintaining the efficiency of a simplified Frequent Pattern Mining approach.

This paper extends a previous preliminary version of the method we proposed in [2], and includes three major novel contributions. The first is a comprehensive review of the state of the art techniques for real-time event detection and trend identification. The second contribution is an in-depth comparison between the performances achieved by our framework and three alternative solutions, namely, a basic SFPM-based approach and two complete real-time systems, i.e. *enBlogue* and *TwitterMonitor*. The third contribution is the validation of the system proposed on data flowed through Twitter during the 64 matches of FIFA World Cup 2014.

The remainder of the paper is organized as follows: some related works are outlined in Section 2. The topic detection system we propose is described in Section 3. Section 4 presents the experimental results. Conclusions are discussed in Section 5.

2. Related Work

The state of the art on topic detection includes several techniques which can be roughly classified into three major categories [1].

The approaches commonly known as *document-pivot* create groups of documents according to a specific document representation and some document-to-document or document-to-cluster similarity measures. For instance, each word in the document can be represented by its value of Term Frequency-Inverse Document Frequency (TF-IDF) [3]. The TF-IDF vectors that represent different documents can be compared to assign a document to an existing cluster, i.e. a topic, or to create a new one. TwitterStand [4] is a news processing system from Twitter tweets which uses TF-IDF and a cosine similarity measures to automatically group news tweets.

Other approaches create clusters of terms by computing the co-occurence patterns between pairs of terms selected among different documents. These methods, called *feature-pivot*, mainly differ to each other by the term selection mechanism they use. For example, the technique proposed in [5] uses parallel

FP-Growth [6] to select frequent word sets from health-related tweets, whilst in [7] emerging terms are selected from Twitter according to both their aging and the *reputation* of the author. Since in many cases considering the co-occurences between *pairs* of terms can be a limitation, *Frequent Pattern Mining (FPM)* approaches are based on the analysis of the co-occurences between any number of terms. A soft version of FPM, called Soft Frequent Pattern Mining (SFPM) [8], is described in more details in section 3.

Other works start from the assumption that some latent topics always exist. The most used *probabilistic topic model* is the Latent Dirichlet Allocation (LDA) [9], where the topic distribution is assumed to have a Dirichlet prior.

The general approaches discussed so far, often represent the basis of several works which analyze data from social networks in order to detect trending topics.

A topic modeling technique based on an online variant of LDA is presented in [10]. The Twitter stream is processed in time slices and older tweets are discarded allowing the model to be constant in size. Moreover, differently from the on-line LDA approach [11], a dynamic vocabulary is maintained in order to deal with new relevant terms detected in emerging topics. Experimental results on synthetic data seem promising, but a comparison against existing approaches is missing.

In [12] a summarization technique for long trending topics is presented. Relevant sub-topics are detected by analyzing the volume of tweets and their semantic (LDA), then the tweets in each subtopic are ranked to generate the sub-summaries. Results show two main limitations which are common to other works: the need for a method to properly determine the number of subtopics and the management of retweets.

The problem of detecting large-scale unexpected events, e.g. earthquakes, is addressed in [13, 14]. The system is designed to monitor users' tweets and a probabilistic spatiotemporal model is built to detect a target event according to specific keywords. Even though the effectiveness on earthquakes detection has been proved, this technique is highly dependent of the query terms, e.g. *earthquake*, *shaking*, and some relevant tweets which do not contain the chosen keywords are ignored.

Other systems try to exploit the explosion of unexpected events to drive the detection process. For example, a framework for real-time detection of *bursty* events is presented in [15], where topics with a sudden increase of popularity are identified by capturing the acceleration of the total number of tweets, the occurrence of words and word pairs.

TwitterMonitor [16] is a framework for Twitter trending topics detection. Trends in topics are modeled by observing the presence of bursty keywords, i.e. keywords whose frequency suddenly changes, and each topic is described by sets of significative keywords. The problem of real-time detection of emergent topics is also addressed in [17]. The authors propose *enBlogue*, a framework based on time-sliding windows to monitor topics that contain set of tags. Correlations between different pairs of tags are analyzed to provide a measure of the onset of new topics. A user-based evaluation on small-scale events shows that *enBlogue* performs better than *TwitterMonitor* [16], however an in-depth assessment of the two systems when dealing with popular events is missing. An in-depth evaluation of enBlogue and TwitterMonitor is presented in section 4.2.

A comparison of different topic detection algorithms discussed in [1] shows that SFPM is a promising solution both for topic detection and representation; we started from these results to design our live topic detection framework.

3. Topic Detection

As previously mentioned, the main advantage of adopting a SFPM-based approach is given by the capability of managing a number of co-occurences greater than two, without requiring that all terms co-occur frequently.

SFPM [8] processes a corpus C of n tweets according to some input parameters, namely the number of *top terms* to be selected (K), and a similarity threshold Θ.

Given a current set of tweets C_{cur}, the first step of the algorithm consists in selecting the K most relevant terms t_k, that is the terms with the highest ratio of the likelihood of appearance in C_{cur} and in a reference corpus C_{ref} of randomly collected tweets:

$$r(t_k) = \frac{p(t_k \mid C_{cur})}{p(t_k \mid C_{ref})}. \tag{1}$$

At the core of SFPM are a set of terms S, which ultimately represents a topic, a vector D_S of i elements, which stores how many of the terms in S co-occur in the i-th document, and a binary vector D_t of i elements, where $D_t(i) = 1$ if the term t occurs in the i-th document.

The set S is expanded by iteratively including the best matching term, i.e. the term t with a cosine similarity with S greater than a threshold Θ, where

$$\Theta(S) = 1 - \frac{1}{1 + e^{\frac{|S|-b}{c}}} \tag{2}$$

is a sigmoid function of $|S|$ that allows to easily add terms to S if its cardinality is low, whilst it becomes more difficult when $|S|$ increases.

The algorithm is repeated K times, being K the number of considered terms, producing topics which may be very similar to each other; for this reason, given two duplicate topics t_1 and t_2, and their similarity value v computed as the percentage of terms common to t_1 and t_2, the smaller topic is deleted if $v > 0.75$.

3.1. Twitter Live Detection

In this section we present our Twitter Live Detection Framework (TLDF), and we show how SFPM, summarized in Algorithm 1, has been modified to meet the constraints of a dynamic detection scenario.

Since relevant Twitter topics rapidly change, a real-time topic detection system must adapt its behavior to the amount of incoming data in order to provide prompt results. A suitable approach to achieve such a goal is to analyze

Algorithm 1 Soft Frequent Pattern Mining

```
function SFPM($C, K, b, c$)
    $T = SFPM\_TermSelection(C, K)$;
    for each term $t$ in $T$ do
        Compute $D_t$;
    for each term $t$ in $T$ do
        $S \leftarrow t$; $D_S \leftarrow D_T$;
        $expand \leftarrow true$;
        repeat
            $t^* \leftarrow BestMatchingTerm(D_S, S, T)$;
            $sim \leftarrow CosineSimilarity(D_S, D_{t^*})$
            if ($sim > \Theta_{b,c}(S)$) then
                $S \leftarrow S \cup t^*$;
                $D_S \leftarrow D_s + D_{t^*}$;
                for i=1 to n do
                    if ($D_{Si} < |S|/2$) then
                        $D_{Si} \leftarrow 0$;
                    else
            else
                $expand \leftarrow false$;
        until expand
        $Topics \leftarrow Topics \cup S$;                    ▷ Initially, $Topics$ is $\emptyset$
    return $RemoveDuplicates(Topics)$;
```

the stream of tweets within meaningful temporal windows selected with a certain criterion.

The method for selecting terms adopted by SFPM allows to identify the most relevant terms in a reference corpus. However, in a dynamic scenario, the current set of terms for the n-th window W_n should depend on the topics detected in the window W_{n-1}. According to Eq.1, this dependency causes existing terms to be generally preferred to emerging terms that are growing in the current window.

In order to prevent such a behavior, the likelihood ratio has been combined with a measure which also stresses the importance of relevant terms in the current set.

Term Frequency-Inverse Document Frequency (TF-IDF) is a function that assigns high scores to those terms with a high frequency in the current set and a low frequency in the whole collection. In particular, given a collection of $|D|$ documents, and a term t that occurs in the document d, the TF-IDF value of t is:

$$TF\text{-}IDF(t) = TF(t,d) \times IDF(t,D), \tag{3}$$

where $TF(t,d)$ is the frequency of the term t in the document d, and

$$IDF(t,D) = log\frac{|D|}{|\{d \in D : t \in d\}|}. \tag{4}$$

Weighting the likelihood ratio $r(t)$ with $TF\text{-}IDF(t)$ allows to filter out common terms and select the terms which are relevant both in the collection (i.e. in the past topics) and in the current window.

Algorithm 2 The proposed term selection mechanism

```
function TLDF_TermSelection($C, K$)
    for each term $t$ in $C$ do
        $p_{new} \leftarrow LikelihoodOfAppearance(t, C_{new})$;
        $p_{ref} \leftarrow LikelihoodOfAppearance(t, C_{ref})$;
        $r_t \leftarrow p_{new}/p_{ref}$;
        $TFIDF_t \leftarrow ComputeTFIDF(t)$;
        if ($NER(t)$) then
            $\omega_t \leftarrow 1.5$;
        else
            $\omega_t \leftarrow 1$;
        $f_t \leftarrow \omega_t \times r_t \times TFIDF_t$;
    $Sort(f, ASCENDING)$;
    for i=1 to K do
        $T \leftarrow T \cup t(f_i)$;                    ▷ Initially, $T$ is $\emptyset$
    return $T$
```

Furthermore, some words (e.g. names of persons) have a level of significance inherently greater than others. In order to deal with this aspect, a Named-Entity Recognition (NER) module [18] has been adopted to test the membership of a certain word to three relevant classes, namely *persons*, *organizations*, *locations*. The importance of the terms detected by the NER process is then boosted by a factor of 1.5 (see [19]).

Thus, the term selection method used by TLDF (see Algorithm 2) chooses the K terms with the highest f-value:

$$f(t) = \omega(t) \times r(t) \times TF\text{-}IDF(t), \tag{5}$$

where $\omega(t) = 1.5$ if t is a named entity recognized by NER, or $\omega(t) = 1$ otherwise.

A further consideration concerns the way in which the size of the detection windows is chosen. Using fixed-size windows is indeed not suitable for real-time topic detection since the actual duration of a topic is generally unpredictable.

In particular, a real-time system must be able to capture both rapid events, which generate a huge amount of tweets in a very short period of time, e.g. a goal in the FIFA world cup final, and long events whose related tweets may go on for several days, e.g. political elections or facts which awaken the public opinion. Such a behavior can be achieved by adopting dynamic detection windows W, whose size depends on the sigmoid function:

$$S(x) = c_1 \left(1 - \frac{1}{1 + e^{-c_2(x-c_4)}}\right) + c_3 \tag{6}$$

where the parameters c_1, c_2, c_3, c_4 control the dynamic range, the slop, the bias, and the centre of the sigmoid respectively [20]. According to Eq. 6, short windows can be used to detect bursty events which involve a huge number of tweets (e.g. $c_1 = 20000$ is the threshold to instantly close a window). As shown in Fig. 1, the trend of the curve changes after 10 minutes ($c_4 = 10$), and the more time elapses, the less tweets are needed to complete a detection window.

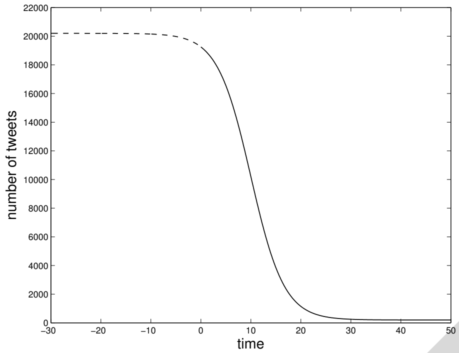


Figure 1: The sigmoid designed to model the behavior of the detection windows.

The slop parameter c_2 has been fixed to 0.3, whilst $c_3 = 200$ in order to capture at least 200 tweets before starting the detection.

Even though the adoption of dynamic windows allows to adapt the behavior of the detector to the actual volume of tweets, a further improvement is required to overcome the limitation of SFPM in detecting unexpected events, e.g. new subtopics, or new trends of the main topic.

We addressed this issue by maintaining a dynamic set of keywords that is continually updated including new terms which reflect the users' perspective on a specific event, or discarding old, unused, words.

More specifically, for the n-th detection window W_n we maintain a list L_n of the most relevant terms in W_n, and a vector of scores I_{tn}, whose values represent the importance of each term t in W_n, computed as the square root of the number of tweets wherein t occurs. The terms with a score above the average are grouped in pairs and added to L_n if at least one of the two terms is trusted, i.e. a named entity recognized by NER module. The life cycle of the new terms is implicitly limited to a single window so as to keep the focus on the event specified by the initial keyword set.

The whole behavior of TLDF is summarized in Fig. 2.

4. Experimental Results

Due to the huge amount of data coming from Twitter, the evaluation of a real-time topic detector results very challenging. Thus, the results of the detection are usually compared to a ground truth [21] obtained by manually labeling each topic as an *event*, if it contains enough information to be related to a real fact, as *spam*, if it does not concern any event, or *neutral*, if it can not be directly related to a specific event.

We moved slightly away from these definitions since, in our perspective, social networks analysis should also consider the *social aspects* of what the users share. Thus, we refuse the presence of *neutral* content since such information is often helpful to discover new trends or topics. Moreover, since we perform

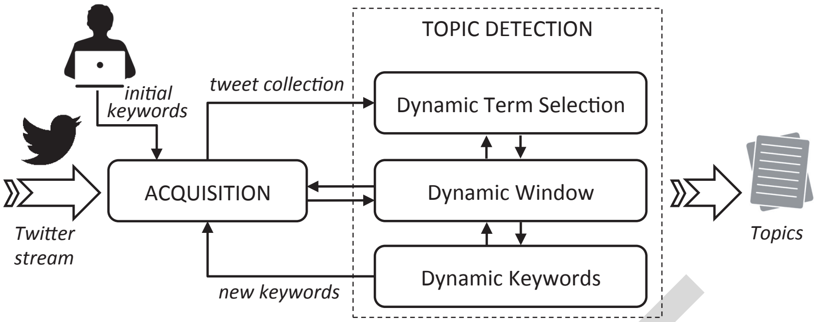


Figure 2: Twitter is queried using a initial set of keywords provided by the user. The acquired tweets are analyzed to select relevant terms. Dynamic detection windows are selected according both to the duration of the event and the number of related tweets. For each window, the set of keywords is updated by including new relevant terms or deleting those unused. A list of topics is obtained.

a keyword-based detection, suitable labels for topics detected in the considered scenario are: *event*, if its keywords are sufficient to understand the related event; *past event*, if its keywords refer to an event already detected in a previous window; *spam*, if its keywords refer to events which are not of interest.

Detected topics were compared to the ground truth in terms of *topic recall*, the percentage of ground truth topics correctly detected; *keyword precision*, the percentage of correctly detected keywords out of the total number of keywords contained in those topics which have been correctly detected in the current window; *keyword recall*, the percentage of correctly detected keywords over the total number of keywords contained in the ground truth topics which have been correctly detected in the current window.

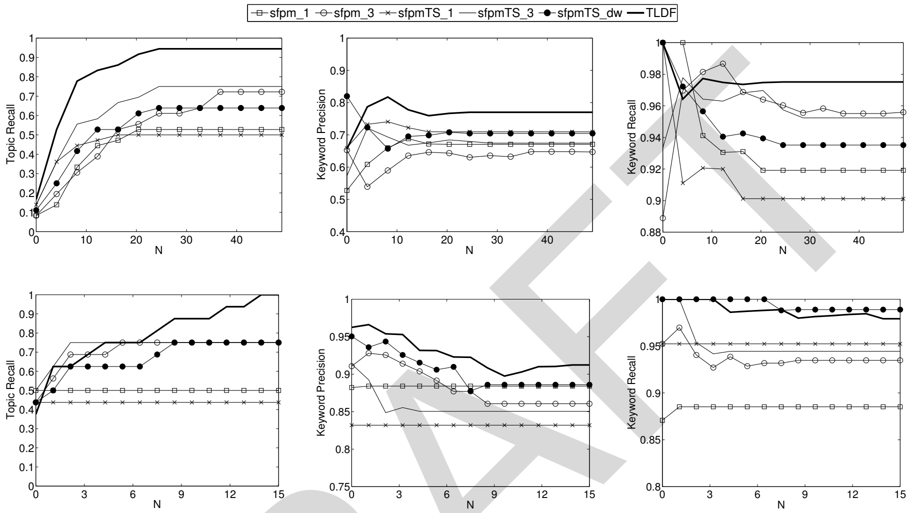
The proposed Twitter Live Detection Framework (TLDF) is intended to be used as an automatic tool for tracking relevant social events from the user's perspective. The detection method is dependent only on the initial set of keywords used to query Twitter, thus it is suitable for very different application scenarios. For example, reporters may be interested in querying Twitter in order to measure the public opinion on a specific event, whilst marketing specialists may want to know how a specific product is accepted by the customers. Being one of the most eagerly-awaited events of 2014, we selected as testing scenario the 64 matches of the FIFA World Cup.

A prototypal version of TLDF has been implemented in Java to facilitate the integration with external components, such as the Twitter4J library [22] used to interact with the Twitter Streaming APIs [23].

In the following of this section we present two different sessions of experiments. The first aims to evaluate the improvements to SFPM discussed in section 3.1, the second is meant for comparing TLDF and two real-time systems, namely enBlogue [17] and TwitterMonitor [16].

sfpm_M	SFPM with M-minute timeslots. [1]
sfpmTS_M	SFPM with the new term selection (TS) algorithm and M-minute timeslots.
sfpmTS_dw	SFPM with the new TS algorithm and dynamic windows.
TLDF	The live detection framework which includes the TS algorithm, dynamic windows and dynamic set of terms.

Table 1: The different configurations used to evaluate the framework.



(a) (b) (c)

Figure 3: Topic Recall (a) Keyword Precision (b) Keyword Recall (c) achieved by the six considered methods for the FIFA World Cup 2014 final (top row) and the match between Australia and Netherlands (bottom row).

4.1. TLDF vs SFPM

In order to analyze the performance of TLDF in terms of *topic recall*, *keyword precision*, and *keyword recall*, we compared our solution with different versions of SFPM (see Table 1). The evaluation is based on the strategy suggested in [8], i.e. by measuring the performances while varying the number of topics N. In order to provide results on events which involved a different number of users, Fig. 3 shows the performances achieved by analyzing the most popular match of the FIFA World Cup 2014, i.e. the final match between Germany and Argentina (top row), and one of the several matches of the first stage of Brazil 2014, i.e. Australia vs Netherlands (bottom row).

Results on the final match show that TLDF achieves the highest performance for all three metrics. The TR values achieved by SFPM with 3-minutes timeslots, i.e. *sfpm_3* and *sfpmTS_3*, are higher than those obtained by *sfpm_1* and *sfpmTS_1*, suggesting that the adoption of windows of 3 minutes is a better choice. Even the use of dynamic windows makes the system to perform bet-

	sfpm	sfpmTS	sfpmTS_dw	TLDF
CPU (%)	25.3	23.7	14.4	**12.6**
RAM (MB)	546	729	480	**517**

Table 2: Average CPU and memory usage of SFPM-based approaches and TLDF.

ter than the static version of SFPM with 1-minutes timeslots. In particular, the average duration of the windows produced by *sfpmTS_dw* is 2.7 minutes, which is near to the duration of the timeslots involved in *sfpm_3* and *sfpmTS_3*. Moreover, regardless of the timeslot durations, the use of the term selection algorithm used in TLDF allows to outperform SFPM proving the effectiveness of the proposed solution.

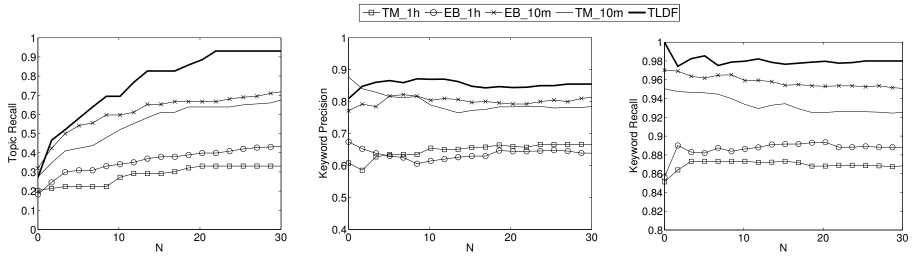
Similar results were obtained during the match between Australia and Netherlands, however, due to the lower number of tweets sent by the users, the average duration of the temporal windows made by *sfpmTS_dw* was 10.3 minutes, whilst *TLDF* used windows of about 12.3 minutes. The performance obtained by these two systems further confirm the effectiveness of adopting dynamic windows to adapt the detection process to the actual volume of tweets. Moreover, the use of a dynamic set of keywords allowed to capture a higher number of tweets more closely related to the considered event.

Tests were performed on a desktop PC with a 2.8GHz dual-core microprocessor, and a comparison of the average CPU and memory usage registered by TLDF and the SFPM-based approaches is reported in Table 2. Results show that the term selection algorithm introduced in *sfpmTS* does not significantly reduce the CPU usage of SFPM, whilst the increase in memory usage is due to use of the NER module. Dynamic windows, i.e. *sfpmTS_dw*, reduce both the CPU and memory usage by allowing the system to process only relevant set of tweets, rather than to force it to analyze fixed size windows every 1-3 minutes. Finally, TLDF maintains the same CPU usage of *sfpmTS_dw*, whilst a bit more RAM is used to manage the dynamic set of terms used to query Twitter.

4.2. TLDF vs enBlogue and TwitterMonitor

Once we compared TLDF with the basic SFPM algorithm, we present here the experiments performed to compare TLDF and two real-time systems. As mentioned in section 2, both enBlogue and TwitterMonitor allow to detect emergent topics by analyzing the correlation between sets of relevant tags. The behavior of enBlogue (EB) can be summarized as follows:

1. **Tag selection:** tags are extracted from tweets by means of a NER process, then at each evaluation stage the most popular tags, called *seed tags*, are chosen. This avoids scalability issues since only the tag pairs that consist of at least one seed tag are used.
2. **Correlation tracking:** tweets that contain the pairs selected at the previous step are monitored to compute the correlation value of two tags,



(a) (b) (c)

Figure 4: Average values of Topic Recall (a) Keyword Precision (b) and Keyword Recall (c) achieved during the last 8 matches of the 2014 FIFA World Cup by TLDF, enBlogue (EB) and TwitterMonitor (TM) with windows of 1 hour and 10 minutes.

that is how important a pair is both locally (how many times two tags appear separately) and globally (how likely it is to see two tags together).

3. **Topic detection:** the correlation values computed at the previous step are analyzed to detect changes which may correspond to new topics.

TwitterMonitor (TM) performs topic detection in two steps by identifying bursty keywords and grouping them according to their co-occurrences.

Both enBlogue and TwitterMonitor use fixed-size windows, thus we tested the two systems with windows of different durations, i.e. 1 hour, as suggested by the authors of [17], and 10 minutes, a time interval more appropriate to the duration of a football match and comparable to the duration of the windows detected by TLDF.

Since the quite small volume of tweets sent during some matches of the early stages of the tournament may affect the performance of coarse-grained systems such as enBlogue and TwitterMonitor, we choose to make a comparison with TLDF by analyzing only the last 8 matches of 2014 FIFA World Cup, i.e. the most discussed ones. Results are shown in Fig. 4.

As expected, EB and TM achieved the worst performances with 1-hour windows. Such a long window causes the misdetection of important topics and an increase of spam because of the very different contents of the analyzed tweets. This result is also confirmed by the values of keyword precision and keyword recall shown in Fig. 4(b) and 4(c). Reducing the size of the windows to 10 minutes allows to obtain better performances and enables EB to perform slightly better than TM. The adoption of windows shorter than 10 minutes (not shown for the sake of Fig. 4 readability) excessively reduces the amount of tweets to be processed and makes both EB and TM unable to detect significant topics. TLDF performs better than its competitors according to the three aforementioned metrics, and this is mainly due to the capability of TLDF to adapt its behavior to the volume and the relevance of the incoming tweets. Moreover, since the analysis is performed at the end of each window, the use of fixed-size

Time	Topic	TLDF	EB	TM
21:00	Khedira is out due to injury.	khedira; injury	-	-
21:30	Higuain goal disallowed for offside.	higuain; offside	higuain; goal	higuain
22:30	Will the game be decided in 90 minutes?	nogoal	overtime	-
22:53	Full-time! Match moves to extra-time.	overtime	-	overtime
23:17	Messi was seen vomiting.	messi; vomiting	-	-

23:24	GOAL! Mario Goetze.	goal;
		goetze;
		germany
			germany;
			goal
				goetze

23:36	Germany are the champions of the world.	germany;
		champions;
			germany;
			cup
				germany;
				wins

23:36	Germany win the World Cup.	germany;
		cup;
		victory
			germany;
			cup
				-

23:37	Gotze goal crowns Germany champions.	germany; champions	-	germany; worldcup
23:58	Germany lift the World Cup.	germany; worldcup	germany; cup	-

Table 3: The 10 most popular topics detected during the FIFA World Cup 2014 final match by means of TLDF, enBlogue (EB) and TwitterMonitor (TM).

windows introduces some delay making making neither EB nor TM suitable for real-time detection of short events, whilst they may be successfully used for long-lasting monitoring of the Twitter stream.

The 10 most popular topics detected by the three systems during the FIFA World Cup final match are reported in Table 3. A noticeable example of the importance of the user's perspective on unexpected events is the topic detected by TLDF at 23:17 CEST, when Messi got sick.

An overall evaluation of the framework was also performed in terms of *precision* and *redundancy* of all the detected topics. The first indicator is somehow connected to the *topic recall*, and is defined as the number of distinct events the system is able to detect, compared to the actual number of distinct events observed during a session. The *redundancy* is the complementary of the number of distinct events the system is able to detect, compared to the number of events detected during a session. Values of *precision* near to 100% indicate the completeness in detecting significative events, whilst the *redundancy* measures the average number of references to the same event; thus, the lower is the value of the *redundancy*, the higher is the generalization capability of the detector.

Table 4 reports a comparison between the average values of precision and redundancy achieved by the proposed live detection technique, enBlogue and TwitterMonitor during the FIFA World Cup 2014 final. Results show that TLDF outperforms its competitors in terms both of precision and redundancy.

Finally, regarding the computational complexity of the considered algorithms, both enBlogue and TwitterMonitor consist of sequences of for loops that take $O(K)$ time, e.g. the analysis of set of K terms. Furthermore, the two

	EB_1h	**TM_1h**	**TM_10m**	**EB_10m**	**TLDF**
Precision	60.4	54.6	75.3	68.7	**92.3**
Redundancy	83.4	87.6	78.2	81.3	**70.4**

Table 4: Average values of Precision (%) and Redundancy (%) achieved during the last 8 matches of the 2014 FIFA World Cup by TLDF, enBlogue (EB) and TwitterMonitor (TM) with windows of 1 hour and 10 minutes.

systems use also a sorting procedure to select the most relevant events according to their score. Thus, if n events are detected, the complexity is $O(K + n \log n)$. On the other hand, the computational complexity of TLDF depends exclusively on the number of terms, K, that are selected. Since the set expansion procedure is repeated K times, and each time goes through the K candidate terms for expansion, the complexity of the algorithm is $O(K^2)$. Such analysis shows that the examined algorithms are comparable since all run in polynomial time. Thus, according to the experimental results discussed so far, TLDF achieves better performances than enBlogue and TwitterMonitor, while consuming a similar amount of computational resources.

5. Conclusion

In this work we presented a framework for real-time analysis of Twitter data in order to detect relevant topics discussed by the users. The analysis of the state of the art suggested us to start from an existing technique, i.e. SFPM, which seemed to provide promising results in offline detection scenarios. Then we designed some improvements to SFPM which allowed to use it for real-time detection of social events.

We run tests on a dataset collected during the FIFA World Cup 2014 and aimed at evaluating the effectiveness of our solution compared with a basic SFPM approach and two real-time systems.

Experimental results using five different metrics (i.e. topic recall, keyword precision, keyword recall, precision and redundancy), showed that our live detection system outperforms other techniques.

Moreover, the most interesting, and quite unexpected, point is that in most cases, other systems were unable to capture the social aspects of the observed events. This happened every time the users left the main topic and started to talk about unexpected events, such as injuries of the players or referee's errors. The detection and tracking of such events has been possible thanks to the dynamic set of keywords we maintain, that allowed to capture new significant topics apparently unrelated to the main event.

As future work we are investigating new text summarization techniques which can speed up the evaluation process by comparing the detected topics with trusted information coming from the Web.

Acknowledgement

This work has been partially supported by the Italian Ministry of Education, University and Research on the "StartUp" call funding the "Bigger Data" project, ref. PAC02L1_0086 CUP: B78F13000700008.

References

[1] L. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, A. Jaimes, Sensing trending topics in Twitter, Multimedia, IEEE Transactions on 15 (6) (2013) 1268–1282.

[2] S. Gaglio, G. Lo Re, M. Morana, Real-time detection of Twitter social events from the user's perspective, in: Communications (ICC), 2015 IEEE International Conference on, 2015, pp. 1–6.

[3] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manage. 24 (5) (1988) 513–523.

[4] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, J. Sperling, TwitterStand: News in tweets, in: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09, ACM, New York, NY, USA, 2009, pp. 42–51.

[5] J. Parker, Y. Wei, A. Yates, O. Frieder, N. Goharian, A framework for detecting public health trends with Twitter, in: Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, 2013, pp. 556–563.

[6] H. Li, Y. Wang, D. Zhang, M. Zhang, E. Y. Chang, Pfp: Parallel fp-growth for query recommendation, in: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08, ACM, New York, NY, USA, 2008, pp. 107–114.

[7] M. Cataldi, L. Di Caro, C. Schifanella, Emerging topic detection on Twitter based on temporal and social terms evaluation, in: Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD '10, ACM, New York, NY, USA, 2010, pp. 4:1–4:10.

[8] G. Petkos, S. Papadopoulos, L. Aiello, R. Skraba, Y. Kompatsiaris, A soft frequent pattern mining approach for textual topic detection, in: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), WIMS '14, ACM, New York, NY, USA, 2014, pp. 25:1–25:10.

[9] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[10] J. H. Lau, N. Collier, T. Baldwin, On-line trend analysis with topic models: #twitter trends detection topic model online, in: COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India, 2012, pp. 1519–1534.

[11] L. AlSumait, D. Barbara, C. Domeniconi, On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking, in: Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on, 2008, pp. 3–12.

[12] D. Gao, W. Li, X. Cai, R. Zhang, Y. Ouyang, Sequential summarization: A full view of Twitter trending topics, Audio, Speech, and Language Processing, IEEE/ACM Transactions on 22 (2) (2014) 293–302.

[13] T. Sakaki, M. Okazaki, Y. Matsuo, Tweet analysis for real-time event detection and earthquake reporting system development, Knowledge and Data Engineering, IEEE Transactions on 25 (4) (2013) 919–931.

[14] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes Twitter users: Real-time event detection by social sensors, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, ACM, New York, NY, USA, 2010, pp. 851–860.

[15] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, K. Wang, Topicsketch: Real-time bursty topic detection from Twitter, in: Data Mining (ICDM), 2013 IEEE 13th International Conference on, 2013, pp. 837–846.

[16] M. Mathioudakis, N. Koudas, Twittermonitor: Trend detection over the Twitter stream, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10, ACM, 2010, pp. 1155–1158.

[17] F. Alvanaki, S. Michel, K. Ramamritham, G. Weikum, See what's enblogue: Real-time emergent topic identification in social media, in: Proceedings of the 15th International Conference on Extending Database Technology, EDBT '12, ACM, 2012, pp. 336–347.

[18] J. R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by gibbs sampling, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 363–370.

[19] S. Phuvipadawat, T. Murata, Breaking news detection and tracking in Twitter, in: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on, Vol. 3, 2010, pp. 120–123.

[20] J. Han, C. Moraga, The influence of the sigmoid function parameters on the speed of backpropagation learning, in: J. Mira, F. Sandoval (Eds.), From Natural to Artificial Neural Computation, Vol. 930 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1995, pp. 195–201.

[21] S. Petrović, M. Osborne, 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, HLT '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 181–189.

[22] Twitter4j - A Java library for the Twitter API.
URL `http://twitter4j.org/`

[23] Twitter Streaming APIs.
URL `http://dev.twitter.com/streaming/`