Real-Time Detection of Twitter Social Events from the User's Perspective

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

In Proceedings of the 2015 IEEE International Conference on Communications (ICC2015)

Publisher: IEEE

Real-Time Detection of Twitter Social Events from the User's Perspective

Salvatore Gaglio, *Member, IEEE,* Giuseppe Lo Re, *Senior Member, IEEE,* and Marco Morana

***Abstract*—Over the last 40 years, automatic solutions to analyze text documents collection have 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 documents, called *tweets*, are usually related to events which involve many people in different parts of the world. In this work we present a system for real-time Twitter data analysis which allows to follow a generic event from the user's point of view. The topic detection algorithm we propose is an improved version of the Soft Frequent Pattern Mining algorithm, designed to deal with dynamic environments. In particular, in order to obtain prompt results, the whole Twitter stream is split in dynamic windows whose size depends both on the volume of tweets and time. Moreover, the set of terms we use to query Twitter is progressively refined to include new relevant keywords which point out the emergence of new subtopics or new trends in the main topic. Tests have been performed to evaluate the performance of the framework and experimental results show the effectiveness of our solution.**

***Index Terms*—Social Sensing, Twitter Analysis, Topic Detection**

I. Introduction

In recent years, the widespread diffusion of social networks has not only allowed people to new forms of interaction within virtual communities, but has also created a new paradigm for sharing information in a pervasive way.

Beside systems which analyze social data to infer new knowledge about user's preferences or activities, recent works are increasingly considering users as sensors able to provide real-time information.

In particular, the most popular social networks, such as Facebook, Twitter, Google Plus+, allow users to share a huge amount of data which reflect their perspective about events which occur all over the world.

We propose here a system to analyze the Twitter stream in order to detect relevant topics within a generic macro event. Differently from other systems 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 to the nature of incoming data.

More specifically, starting from the choice of some generic terms to listen to on Twitter, we split the stream of tweets into dynamics windows which are progressively analyzed to detect relevant topics. As time passes, the initial set of keywords is updated to include new important terms emerging from the tweets themselves, or to delete those unused. The core of the topic detection algorithm is an improved version of Soft Frequent Pattern Mining (SFPM), designed to overcome the limitation of SFPM in dealing with dynamic, real-time, scenarios.

Experimental results on the tweets posted during the 64 matches of FIFA World Cup 2014 show the effectiveness of the proposed solutions.

The remainder of the paper is organized as follows: some related works are outlined in Section II. The topic detection system we propose is described in Section III. Section IV presents the experimental results. Conclusions are discussed in Section V.

II. Related Work

Due to the huge number of topic detection techniques available in literature, it is reasonable to consider the existing methods as belonging to three major categories [1].

Some approaches, generally called *document-pivot*, group documents into clusters according to a specific document representation and some document-to-document or document-to-cluster similarity measures. A common document representation is that based on the Term Frequency-Inverse Document Frequency (TF-IDF) weighting function [2]. For each word in the document, the TF-IDF value increases proportionally to the number of times that word appears in the document, but is offset by the frequency of the word in the collection of documents. TF-IDF vectors representing different documents can be compared to assign a document to an existing cluster, i.e., topic, or to create a new one. For instance, TwitterStand [3], a news processing system from Twitter tweets, uses TF-IDF and a cosine similarity measures to automatically group news tweets.

Differently from *document-pivot* methods, where documents are grouped according to their similarity, *feature-pivot* approaches create clusters of terms by computing the co-occurence patterns between pairs of terms selected among different documents. Feature-pivot methods mainly differ to each other by the term selection mechanism they use. For example, in [4] emerging terms are selected from Twitter according to both their aging and the *authority* of the users which posted them, whilst in [5] parallel FP-Growth [6] is used to select frequent word sets from health-related tweets.

Frequent Pattern Mining (FPM) approaches [7] overcome the limitations of feature-pivot techniques by considering the

S. Gaglio and G. Lo Re and M. Morana are with the DICGIM, University of Palermo, Viale delle delle Scienze, Ed. 6, 90128, Palermo - ITALY

S. Gaglio is also with ICAR-CNR, National Research Council of Italy, Viale delle delle Scienze, Ed. 11, 90128, Palermo - ITALY

co-occurences between any number of terms. This generally improves the quality of the detected topics, but efficient algorithms are needed to discover frequent patterns in all possible set of terms. A soft version of FPM, called Soft Frequent Pattern Mining (SFPM) [8], is described in more detail in section III.

Other approaches, called *probabilistic topic models* are based on the assumption that some latent topics always exist; thus, each document can be considered as a mixture of different latent topics. The most used probabilistic 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, have been adopted in several works which analyze social data for different purposes.

A framework for real-time detection of *bursty* events is presented in [10]. Topics with a sudden surge of popularity are modeled by *sketches* which capture the acceleration of the total number of tweets, the occurrence of words and word pairs. Hashing techniques are applied to efficiently maintain data sketch from which bursty topics are inferred.

In [11] the problem of summarizing long trending topics is discussed. The idea is to generate chronologically related sub-summaries which cover the entire development of the topic. Firstly, two detectors, based on the volume of tweets and their semantic (LDA), are used to identify relevant subtopics, 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, requested by probabilistic detectors, and the management of retweets, which provide a measure of how important a tweet is but may also introduce noise to the detection process.

A real-time event detector is presented in [12], [13]. The system is designed to monitor users' tweets and a probabilistic spatiotemporal model is built to detect a target event according to specific keywords. Results on earthquake detection show the effectiveness of this solution, however this approach is too highly dependent of the query terms, e.g., *earthquake*, *shaking*, and some relevant tweets which do not contain the chosen keywords are ignored. Moreover, the assumption that a single instance of the target event exists, i.e., two earthquakes do not occur simultaneously, is not acceptable for any type of event.

In [14] a technique to generate summaries of a sporting event using only tweets is presented. Firstly, the salient points of an event are detected by observing the spikes in the number of related tweets. Then the most relevant sentences within those points are detected by means of a phrase graph ranking algorithm [15] and used to build the summary. Even if this method provide good results in terms of readability, the main limitation is that the summary contains the same words of the tweets and so its completeness depends on that of the tweets.

A comparison of different topic detection algorithms (i.e., document-pivot, LDA, feature-pivot, FPM, SFPM, BNgram) tested on Twitter data is presented in [1]. Results show that SFPM is a promising solution both for topic detection and representation; thus, we started from SFPM to design our live detection system.

III. Topic Detection Framework

As mentioned in section II, the most important characteristic of *feature-pivot* methods is the evaluation of *pairs* of terms to create clusters of correlated keywords. This idea is extended by *frequent pattern mining* techniques which search for co-occurences between any number of terms instead.

Soft Frequent Pattern Mining (SFPM) [8] lies between these two approaches by considering a number of co-occurences, P, greater than two, but not strictly requiring that all P terms co-occur frequently.

In order to produce a set of topics, SFPM requires a corpus C of n documents, i.e., tweets, a value K which specifies the number of *top terms* to be selected and two parameters b, c used to compute a similarity threshold Θ.

The fist step of SFPM consists in choosing the K most relevant terms in the current set of tweets C_{cur}, i.e., those related to the most discussed topics. This task is accomplished by selecting the K terms t_k with the highest ratio of the likelihood of appearance in C_{cur} and in a reference corpus C_{ref} of randomly collected tweets:

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

Once that K terms have been selected, the SFPM algorithm maintains 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.

A greedy approach is used to expand the set S by selecting the best matching term, i.e., the term t with a cosine similarity with S greater than a threshold Θ. Such a threshold is computed as a sigmoid function of $|S|$:

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

so that it is easier to add terms to the set S if it contains a few terms, but it is harder and harder when its cardinality 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 and their similarity value v, computed as the percentage of shared terms, the smaller topic is deleted if $v > 75\%$.

The SFPM algorithm is summarized in Algorithm 1.

A. Live Twitter Topic Detection

In this section we provide a description of the mechanisms we designed to adapt the SFPM algorithm to the considered live detection scenario.

In a real scenario, where relevant topics may rapidly change, the topic detection system must adapt its behavior to the amount of incoming data in order to guarantee prompt, i.e., real-time, results. This can be achieved by splitting the huge stream of tweets in distinct temporal windows, selected with a certain criterion, so to perform a more reliable topic detection within a single, meaningful, window.

Algorithm 1 *SFPM (C, K, b, c)*

```
$T = SFPM\_TermSelection(C, K)$;
for each term $t$ in $T$ do
    Compute $D_t$;
end for
$Topics = \emptyset$;
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
                end if
            end for
        else
            $expand \leftarrow false$;
        end if
    until expand
    $Topics \leftarrow Topics \cup S$;
end for
return $RemoveDuplicates(Topics)$
```

The core of SFPM is the term selection method which allows to identify the most relevant terms in a reference corpus. However, in a dynamic scenario, the initial set of terms for the windows W_n depends on the topics detected in the window W_{n-1}; thus, according to (1), existing terms will be generally preferred to those whose are emerging in the current window.

To prevent this behavior, we decided to combine the likelihood ratio with a measure which also gives importance to the relevant terms in the current set.

Term Frequency-Inverse Document Frequency (TF-IDF), briefly presented in section II, is a weighting function which tends to filter out common terms. In particular, a high TF-IDF scores is reached by those terms with a high frequency in the current set and a low frequency in the whole collection.

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 select the terms which are relevant both in the collection and in the current corpus, that is the terms related to the existing topics and those whose are emerging in the current window.

Moreover, some words, such as the names of persons, organizations, locations, are implicitly more important than others in everyday life. For this reason we adopted a Named-Entity Recognition (NER) module [16] to test if a term belongs to one of three relevant classes (*persons*, *organizations*, *locations*) and then boost its importance by a factor of 1.5 (see [17]).

Algorithm 2 *TermSelection (C, K)*

```
$T = \emptyset$;
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
        $\omega_t \leftarrow 1$;
    end if
    $f_t \leftarrow \omega_t \times r_t \times TFIDF_t$;
end for
$Sort(f, ASCENDING)$;
for i=1 to K do
    $T \leftarrow T \cup t(f_i)$;
end for
return $T$
```

Thus, the proposed term selection method, described in 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.

The size of the detection windows W usually depends on the duration of the event you want to observe. For example, in [1], 10 minutes windows were used for two-day long events (U.S. Super Tuesday and U.S. Presidential Election), whilst a short football match (the FA Cup final) was split into 2 minutes timeslots.

This approach is not suitable to be applied for real-time detection of topics whose duration is unpredictable.

Since the importance of a topic is not only dependent of its duration, but is also strictly connected to the number of related tweets, our perspective is to consider dynamic windows whose size depends on both aspects.

In order to design an effective mechanism to control the behavior of such windows, we observed some real events focusing on the relationships between the amount of tweets and the number of timeslots, and the correspondences between detected topics and real events. Results suggested that 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.

This behavior can be reached by means of a sigmoid function [18]:

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

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. In particular, short windows are used to detect bursty

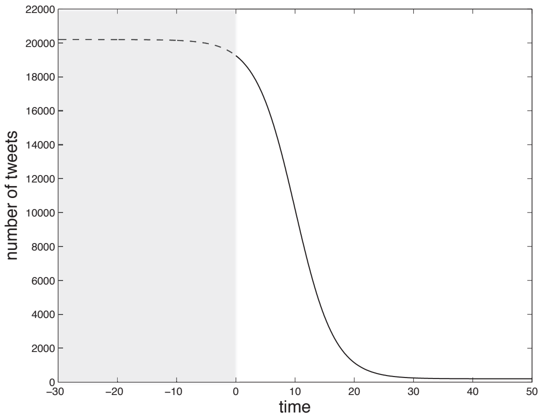


Fig. 1: The sigmoid function designed to model the behavior of the detection windows. The grey area (negative time) is plot to show the whole trend of the curve.

events which involve a huge number of tweets ($c_1 = 20000$ is the threshold to instantly close a window). The inflection point is reached after 10 minutes ($c_4 = 10$), then the trend of the curve changes and the more time elapses, the less tweets are needed to complete a detection window (see Fig. 1). The slop parameter is $c_2 = 0.3$, whilst c_3 is set to 200 in order to capture at least 200 tweets before starting the detection.

Using dynamic windows allows to adapt the behavior of the detector to the actual volume of tweets for a given event. However, querying the stream by a fixed set of terms frequently leads to miss unexpected events, e.g., new subtopics, or new trends of the main topic. We deal with this limitation by means of a controller which is responsible for updating the set of keywords, including new terms which reflect the users' perspective on a specific event or deleting those unused.

More specifically, for each 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, of which at least one of the two terms is required to be a named entity recognized by NER, and added to L_n. The selection of pairs of relevant terms helps to prevent the application of generic filters which may increase the noise level within each window, whilst named entities may support the detection of new topics being trusted terms.

Please note that the life cycle of the new terms is implicitly limited to a single window. For each window W_n, the controller selects those terms whose have been really significant in W_{n-1}; thus new terms which actually refers to emerging topics will be confirmed, whilst others omitted. This also guarantees to keep the focus on the event specified by the initial keyword set, which better reflects the user's intention.

IV. Experimental Results

The evaluation of a real-time topic detection system is a tricky process due to the huge amount of information you have to manage.

TABLE I: The six configurations used to evaluate the framework.

sfpm_1	SFPM with 1-minute timeslots. [1]
sfpm_3	SFPM with 3-minutes timeslots.
sfpmTS_1	SFPM with the new term selection (TS) algorithm and 1-minute timeslots.
sfpmTS_3	SFPM with the TS algorithm and 3-minutes timeslots.
sfpmTS_dw	SFPM with the TS algorithm and dynamic windows.
sfpm_LD	The live detection system which includes the TS algorithm, dynamic windows and dynamic set of terms.

To the best of our knowledge, a fully automatic evaluation protocol does not exist. For this reason, the detected topics are usually compared to a given ground truth where any documents in the collection are manually marked as: *event* (a self-contained text which contains enough information to be related to a real fact), *neutral* (a text which can not be directly related to a specific event), *spam* (a text which does not concern any event)

In our perspective, a system designed to detect topics in social networks should also consider the *social aspects* of what the user decided to share. Thus, we do not consider *neutral* tweets since they may even contain useful information which can be used to discover new trends or topics. Moreover, since the whole detection process is based on keywords, we introduce the following definitions:

- *event*: a topic whose keywords are sufficient to understand the related event;
- *spam*: a topic whose keywords refer to events which are not of interest;
- *past event*: a topic whose keywords refer to an event already detected in a previous window.

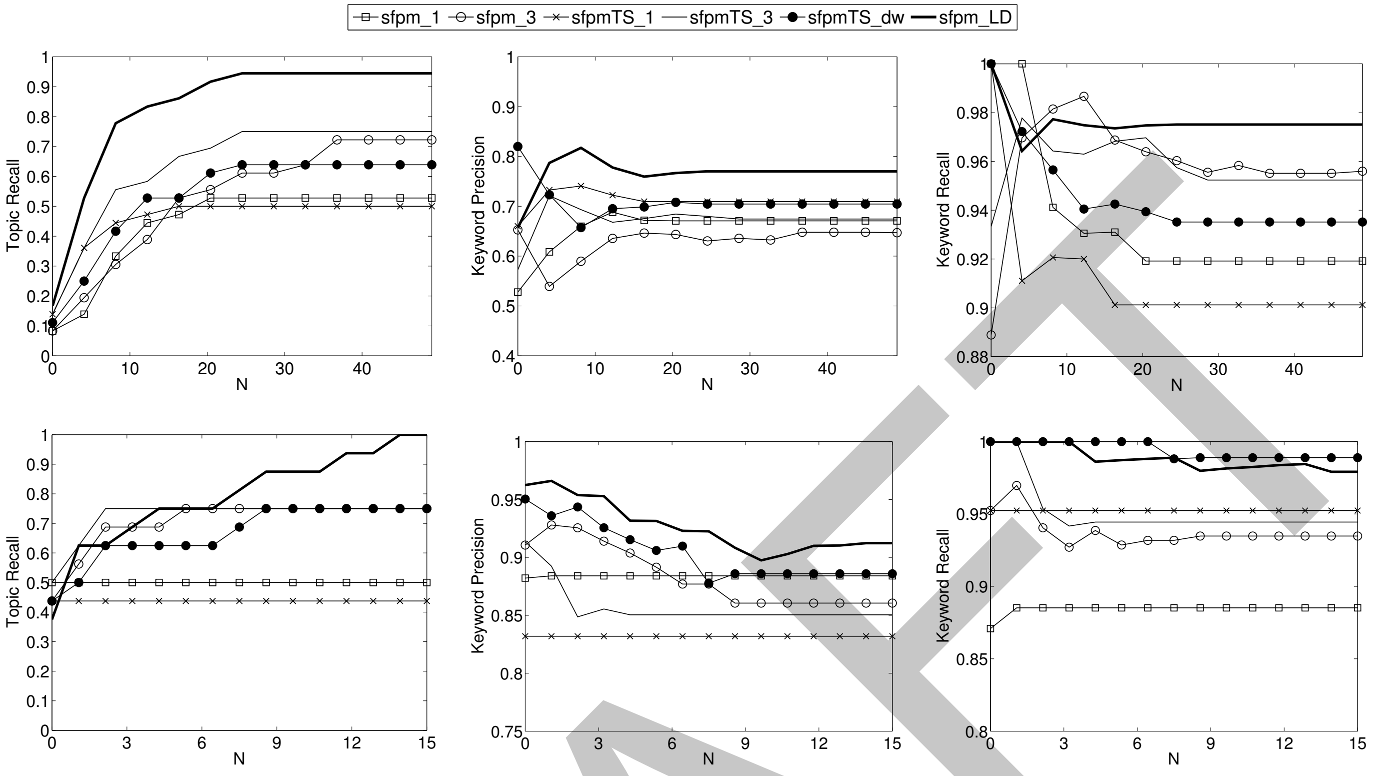
To evaluate the performance of the framework, the detected topics were compared to the ground truth in terms of *topic recall*, *keyword precision*, and *keyword recall*:

- *topic recall*: percentage of ground truth topics correctly detected;
- *keyword precision*; 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*: 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.

Being one of the most eagerly-awaited events of 2014, we selected as testing scenario the 64 matches of the FIFA World Cup. In order to verify the contribution of each of the proposed solution separately, six different systems were compared as shown in Table I.

To provide a significative example of the achieved performance, we present here the results obtained by analyzing the most popular match of the FIFA World Cup 2014, i.e., the final between Germany and Argentina.

The set of keywords used to query the Twitter stream for this event was: *brasil2014, brazil, brasil, worldcup2014, worldcup, world cup, FIFAWC2014, ARGVsGER, GERVsARG*. The same



(a) (b) (c)

Fig. 2: 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).

set was employed both for the static versions of *sfpm*, and as initial set for *sfpm_LD*. The data collection process started 4 hours before the beginning of the match; Fig. 3 shows the volume of tweets captured from the kick off (at 9 p.m.) to the end of the match (156 minutes after).

In order to analyze the three performance metrics above described, we adopted the approach proposed in [8], which consists in the evaluation of the six methods of Table I while varying the number of topics. In particular, the first row of Fig. 2 shows the Topic Recall (TR), Keyword Precision (KP) and Keyword Recall (KR) achieved during Germany-Argentina by considering the top N detected topics.

The proposed live detection system, i.e., *sfpm_LD*, achieves the highest performance for all three metrics. The TR values achieved by the methods 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 better 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 new term selection algorithm allows to outperform *sfpm* proving the effectiveness of the proposed solution.

To further support our analysis we also show the results obtained on one of the many matches of the first stage of Brazil 2014, namely that between Australia and Netherlands played on Jun 18, 2014. The TR, KP and KR values achieved for this event are shown in the bottom row of Fig. 2. Due to the minor number of tweets, the average duration of the temporal windows made by *sfpmTS_dw* was 10.3 minutes, whilst the live detector *sfpm_LD* 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.

Finally, note that since we used the Twitter Streaming API, which allows to get up to 1% of the total firehose, many of the 32.1 million tweets sent by the users during the final were lost. For this reason the traffic shown in Fig. 3 is almost flat, not directly reflecting any significant event. Conversely, the spikes of tweets related to Australia VS Netherlands (Fig. 4) reflect the most important events occurred during the match. In particular, 5 goals were scored after 20', 21', 69', 74' and 83' from the kick off. This trend shows that relying only on the volume of tweets could not be effective when dealing with very popular events.

V. Conclusion

In this work we presented a system to perform real-time analysis of Twitter data. We started from an existing technique, i.e., SFPM, which usually provides good results in offline

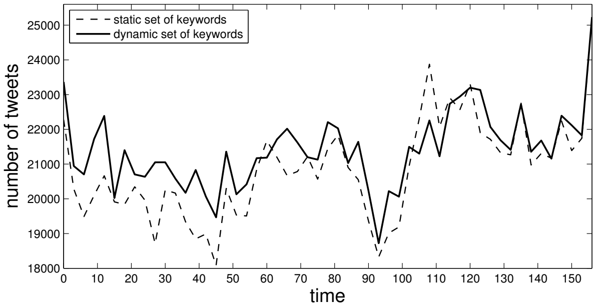


Fig. 3: Comparison of the number of tweets captured during the FIFA World Cup 2014 final using a static and a dynamic set of keywords.

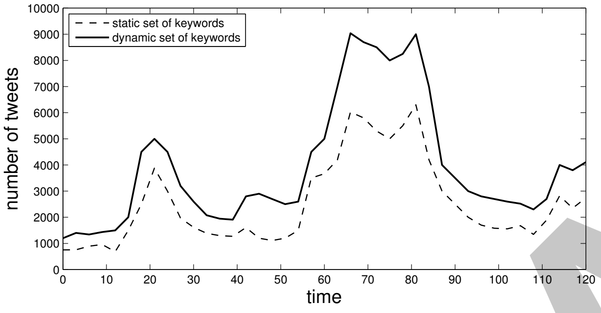


Fig. 4: Comparison of the number of tweets captured during the match between Australia and Netherlands using a static and a dynamic set of keywords.

detection scenarios. Then we proposed some modifications to adopt SFPM for real-time detection of social events. In particular, we defined a new term selection algorithm which allows to select not only the terms related to the existing topics, but also those whose are emerging in the current detection window. In order to capture both rapid and long events, we designed a function to dynamically adapt the behavior of the detector to the actual volume of tweets, that is both to the number of tweets and the duration of the topic. Finally, we presented a mechanism to update the set of keywords used to query Twitter, including new terms which reflect the users' perspective on a specific event or deleting those unused. Several tests were performed during the FIFA World Cup 2014 to evaluate the effectiveness of such solutions.

Experimental results showed that our live detection system outperforms SFPM improving the detection in terms of topic recall, keyword precision and keyword recall. In most cases, only the proposed live detection system has been able to capture the social aspects of the events and this mainly happened when users left the main topic and started to talk about unexpected events, e.g., injuries of the players, demonstrations near the stadium, referee's errors. Tracking such events has been possible thanks to the maintenance of a dynamic set of keywords, which allowed to capture new significant topics apparently unrelated to the main event.

Future work can concern the improvement of the evaluation process by means of trusted information, coming from the Web, to automatically use as ground truth.

References

[1] L. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, and A. Jaimes, "Sensing trending topics in twitter," *Multimedia, IEEE Transactions on*, vol. 15, no. 6, pp. 1268–1282, Oct 2013.

[2] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, Aug. 1988.

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

[4] M. Cataldi, L. Di Caro, and 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*, ser. MDMKDD '10. New York, NY, USA: ACM, 2010, pp. 4:1–4:10.

[5] J. Parker, Y. Wei, A. Yates, O. Frieder, and 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*, Aug 2013, pp. 556–563.

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

[7] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 55–86, 2007.

[8] G. Petkos, S. Papadopoulos, L. Aiello, R. Skraba, and 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)*, ser. WIMS '14. New York, NY, USA: ACM, 2014, pp. 25:1–25:10.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

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

[11] D. Gao, W. Li, X. Cai, R. Zhang, and Y. Ouyang, "Sequential summarization: A full view of twitter trending topics," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 2, pp. 293–302, Feb 2014.

[12] T. Sakaki, M. Okazaki, and Y. Matsuo, "Tweet analysis for real-time event detection and earthquake reporting system development," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 4, pp. 919–931, April 2013.

[13] ——, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 851–860.

[14] J. Nichols, J. Mahmud, and C. Drews, "Summarizing sporting events using twitter," in *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*, ser. IUI '12. New York, NY, USA: ACM, 2012, pp. 189–198.

[15] B. Sharifi, M.-A. Hutton, and J. Kalita, "Summarizing microblogs automatically," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 685–688.

[16] J. R. Finkel, T. Grenager, and 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*, ser. ACL '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 363–370.

[17] S. Phuvipadawat and 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, Aug 2010, pp. 120–123.

[18] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *From Natural to Artificial Neural Computation*, ser. Lecture Notes in Computer Science, J. Mira and F. Sandoval, Eds. Springer Berlin Heidelberg, 1995, vol. 930, pp. 195–201.