An Optimization Approach for Sub-event Detection and Summarization in Twitter

Polykarpos Meladianos[1,2(✉)], Christos Xypolopoulos[1], Giannis Nikolentzos[1,2], and Michalis Vazirgiannis[1,2]

[1] Lix, École Polytechnique, Palaiseau, France
christos.xypolopoulos@polytechnique.edu
[2] Athens University of Economics and Business, Athens, Greece
{pmeladianos,nikolentzos,mvazirg}@aueb.gr

Abstract. In this paper, we present a system that generates real-time summaries of events using only posts collected from Twitter. The system both identifies important moments within the event and generates a corresponding textual description. First, the set of tweets posted in a short time interval is represented as a weighted graph-of-words. To identify important moments within an event, the system detects rapid changes in the graphs' edge weights using a convex optimization formulation. The system then extracts a few tweets that best describe the chain of interesting occurrences in the event using a greedy algorithm that maximizes a nondecreasing submodular function. Through extensive experiments on real-world sporting events, we show that the proposed system can effectively capture the sub-events, and that it clearly outperforms the dominant sub-event detection method.

1 Introduction

Twitter is a very popular microblogging service that allows users to post real-time messages known as tweets. Due to its instantaneous nature, Twitter has been established as a major communication medium. Among others, people use the service to report latest news and to comment about real-world events [6]. Users show particular interest in social events such as large parties, political campaigns and sporting events but also for emergency events such as natural disasters and terrorist attacks [3]. Tweets posted by people involved in these events could provide different perspectives regarding the events compared to the ones that appear in traditional media. Besides the above mentioned types of events, a plethora of other events are reported daily on Twitter, the majority of which are not covered systematically by traditional media. An extreme example of such events are those related to the personal wellness of Twitter users [1].

Users are often interested in tracking the evolution of an event that spans a time interval, such as a football match. Most events typically consist of a sequence of important moments or *sub-events* which attract the attention of users. However, due to the massive pace of generated data, accurately detecting

G. Pasi et al. (Eds.): ECIR 2018, LNCS 10772, pp. 481–493, 2018.
https://doi.org/10.1007/978-3-319-76941-7_36

all sub-events of an evolving event is a very challenging task. Besides the identification of the important moments, it is necessary to provide users with a textual description of the events that are being reported on Twitter. The problem of generating a summary for an event using Twitter is related to the problem of multi-document summarization which has been studied extensively in the past [15]. However, instead of a static collection of well-formatted documents, in the Twitter setting, there exists a stream of tweets with noisy content posted by heterogeneous users which makes the summarization task very hard.

From the above, it is clear that event summarization in Twitter can be seen as consisting of two parts: (1) a sub-event detection mechanism capable of identifying the important moments within an event, and (2) a text generating module which creates text descriptions that best summarize a given sub-event. Both these parts have to be tailored to the particularities of microblogging content. In this paper, we propose a novel system that deals with both aforementioned challenges of event summarization in Twitter. Our system decomposes events into time intervals and represents the set of tweets posted during each time interval as a graph. We assume that important moments within an event trigger a rapid change in the vocabulary employed by users and consequently rapid changes in edge weights. To detect important moments within an event, the system uses a convex optimization formulation which accurately determines the amount of change in the edge weights between the current time interval and the previous time intervals. Given the successful detection of an important moment, to generate a textual description, we propose optimizing a submodular function which takes into account the weights of the edges connecting terms of the tweets that have been added into the summary. The function encourages the produced summary to be both representative of the set of tweets and at the same time diverse. The source code of the proposed system is publicly available[1].

The rest of this paper is organized as follows. Section 2 provides an overview of the related work and elaborates our contribution. Section 3 provides a detailed description of our proposed system. Section 4 evaluates the proposed approach. Finally, Sect. 5 summarizes the work.

2 Related Work

Although much effort has been devoted to the problem of event detection in Twitter [2,18,22,23,25], considerably less attention has been paid to the problems of sub-event detection and summarization.

Most existing systems assume that a sharp increase in the volume of status updates corresponds to the occurrence of an important moment within the event. Existing systems employ different approaches to identify such sharp increases in the tweet rate [16,24,26]. To identify important moments, Chierichetti et al. made also use of the retweet rate [5]. Chakrabarti and Punera proposed in [4] a system that does not depend solely on the tweet/retweet rate. Instead, the authors used a modified Hidden Markov Model which detects sub-events based

[1] https://bitbucket.org/ksipos/optimization-sub-event-detection.

on both the tweet rate and the word distribution used in tweets. Shen et al. identified the participants of events and used a mixture model to detect sub-events for each participant [20]. Srijith et al. proposed in [21] a sub-event detection approach that is based on hierarchical Dirichlet processes, a probabilistic topic model which can learn latent sub-stories associated with tweets. The work closest to ours is perhaps the one reported in [11]. The authors represent sequences of tweets as graphs and sub-events are identified using the notion of graph degeneracy. In our work, we also build graphs to represent sequences of tweets. However, in contrast to the above work, we propose solving an optimization problem to identify the important moments. Furthermore, in contrast to all existing systems, we design and optimize a novel submodular function in order to generate a summary for each sub-event. The work of Letsios et al. is also related to the proposed approach, but it focuses on the problem of event detection [7]. As regards the summarization task, to extract a representative sentence for each important moment, most systems employed the *tf-idf*-based and graph-based approaches proposed by Sharifi et al. [19]. Mackie et al. compared in [10] several methods for Twitter summarization. The TREC Real-Time Summarization Track has been recently created to foster the development of systems that automatically monitor streams of social media posts to keep users up to date on topics of interest [9].

3 Sub-Event Detection and Summarization in Twitter

In this paper, we developed a system for generating real-time summaries of events by using solely status updates collected from Twitter. The system is composed of several modules, and is illustrated in Fig. 1. In what follows, we present the different modules of the proposed system.

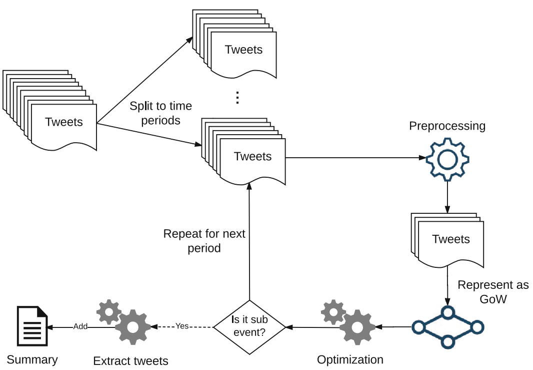

Fig. 1. Overview of our proposed real-time sub-event detection and summarization system.

3.1 Data Preprocessing

In this Section, we give details about the preprocessing steps followed. Data harvested from social media is often noisy and heterogeneous by nature. Taking also into account that social media platforms including Twitter have been infiltrated by various types of unwanted content, such as spam, advertisements and malicious content, it becomes obvious that data preprocessing is a task that should be considered with care.

Given a set of raw tweets $\mathcal{D}_{raw} = \{d_1, d_2, \ldots, d_N\}$, we first remove all retweets and duplicate posts, as they just reproduce the content of other tweets and do not provide additional information. Note that retweets and duplicate tweets constitute a large fraction of the set of tweets $\mathcal{D}_{raw}$. By eliminating them, the proposed system gains both in terms of running time and in terms of performance as these posts increase the noise levels that the system is exposed to. Furthermore, we removed tweets containing "@"-mentions as we assume that in most cases these tweets are not relevant to the event under consideration. All remaining tweets undergo standard text processing tasks including (1) tokenization, (2) stopword removal, (3) punctuation and special character removal, (4) URL removal, and (5) stemming using Porter's algorithm. The preprocessed tweets are then transformed into graphs as described below.

3.2 Graph-of-words Representation

Given the set of preprocessed tweets $\mathcal{D}$, we represent each tweet as a statistical *graph-of-words*, following earlier approaches in keyword extraction [12,13], in summarization [11], and in text categorization [17].

More formally, given the set $\mathcal{D}$ of preprocessed tweets, each tweet corresponds to a sequence of terms. From this sequence we create a graph whose vertices correspond to the unique terms of the tweet. An edge is then drawn between all pairs of vertices of the graph. Hence, a tweet can be seen as a fully-connected graph-of-words (i. e. clique). We chose to link a vertex with all the other vertices instead of a subset of them because the length of a tweet is very short (at most 140 characters). As regards the weights of the edges, we followed an earlier approach and we considered that each term co-occurrence in the tweet is equally important [11]. Therefore, the weight of each edge of the graph is set equal to $1/(n-1)$ where n is the number of unique terms in the tweet. Following this approach, the degree of all vertices is equal to 1.

After transforming the tweets into graphs, we create a single graph G_i that corresponds to the time period i. Let $\mathcal{G}_i$ be a set containing all the graph-of-words representations of the tweets posted during time period i. First, graph G_i is initialized. At this point, G_i is an empty graph. The graphs of $\mathcal{G}_i$ (i. e. graph representations of tweets) are then added sequentially into G_i. Any vertices and edges of these graphs that are not contained in G_i are added to it, while the weights of existing edges are increased by the corresponding weights in these graphs. Hence, pairs of terms that are repeated in many tweets are expected to have a high edge weight between them. Figure 2 illustrates the graph-of-words

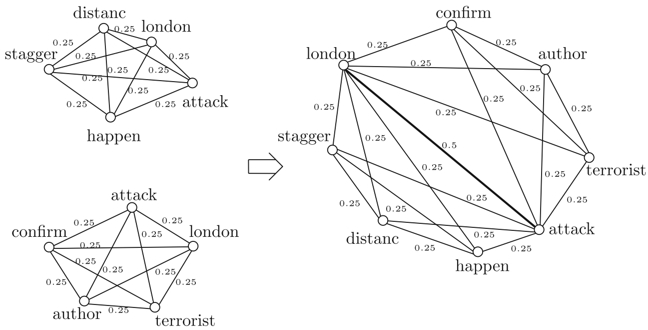

Fig. 2. The graph-of-words representations of two tweets (left), and the single graph that emerges from these two tweets (right).

representations of the following two tweets (left): (1) "*The distance over which the #London attack happened is staggering*", and (2) "*Confirmed by authorities terrorist attack in London*". Furthermore, it shows the graph that emerges after merging the two graph-of-words (right).

3.3 Sub-event Detection

This Section describes our proposed system for detecting important moments within an event from a sequence of graphs-of-words. Let $\mathcal{D}_t$ be the set of tweets posted at time period t, and G_t the graph constructed from this set as described above. Let also i be the current time period. We denote the number of unique terms in the set of tweets $\mathcal{D}_i$ by n.

We propose a novel approach which identifies (1) how much the content of the messages posted by users at the current time period i has changed compared to the previous time periods, and (2) if there are any pairs of words whose quantity of appearances was abnormally high. In such a case, the system considers that an important moment has occurred at time period i and the next module is activated to provide the user with a textual description of the corresponding sub-event. The proposed system identifies such interesting occurrences by solving a convex optimization problem. In what follows, we give details about how this optimization problem is formulated.

The first step is to transform graph G_i corresponding to the current time period i into a vector $\mathbf{b} \in \mathbb{R}^{n^2}$. Specifically, vector $\mathbf{b}$ is created by applying the $vec(\cdot)$ operator to the adjacency matrix of G_i. The $vec(\cdot)$ operator creates a column vector from a matrix by stacking the column vectors of the matrix below one another. Therefore, if $\mathbf{A}_i \in \mathbb{R}^{n \times n}$ is the adjacency matrix of G_i, then

$$\mathbf{A}_i = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \mathbf{b} = vec(\mathbf{A}_i) = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}$$

Then, given the graph representations of the tweets created at the last p time periods $G_{i-p}, \dots, G_{i-1}$, we also transform them into vectors using again the $vec(\cdot)$ operator. However, given each of these graphs, we do not utilize its own adjacency matrix as input to the $vec(\cdot)$ operator, but instead a matrix whose rows and columns correspond to these of the adjacency matrix $\mathbf{A}_i$ of G_i and whose values are set according to the graph under consideration. Hence, the dimensionality of these vectors (as in the case of $\mathbf{b}$) is also equal to n^2. Finally, we construct a matrix $\mathbf{W} \in \mathbb{R}^{n^2 \times p}$ having these vectors as columns

$$\mathbf{W} = \begin{bmatrix} | & & | \\ \mathbf{w}_{i-p} & \dots & \mathbf{w}_{i-1} \\ | & & | \end{bmatrix}$$

In other words, for each one of the previous p periods, we extract the weights of the edges connecting all pairs of vertices (for pairs of vertices that are not connected we assume zero weight) of the graph G_i created from tweets posted at the current period and we create a vector $\mathbf{w} \in \mathbb{R}^{n^2}$. The weights are placed in exactly the same order as in vector $\mathbf{b}$. Hence, entries with the same index correspond to the same pair of vertices (i. e. terms). Then, matrix $\mathbf{W}$ is simply a matrix with columns the vectors $\mathbf{w}_{i-p}, \dots, \mathbf{w}_{i-1}$.

The proposed optimization problem is then formulated as follows

$$\begin{aligned} \min_{\mathbf{x}} \frac{1}{2} &||\mathbf{W}\mathbf{x} - \mathbf{b}||_2^2 \\ \text{s.t.} \quad & \mathbf{1}^\top \mathbf{x} = 1 \\ & x_i \geq 0, \quad \forall i = 1, \dots, p \end{aligned} \tag{1}$$

Let $\mathbf{x}^* \in \mathbb{R}^p$ be the solution of the optimization problem (1). To a certain extent, the values of $\mathbf{x}^*$ indicate how similar is the content of the tweets of the current time period when compared to each one of the previous p periods. For example, if at one of the previous periods users posted tweets describing the same aspect of the event as tweets posted at the current time period, the edge weights corresponding to the same pairs of vertices in the two graphs will have almost equal values. In such a case, the entry of $\mathbf{x}^*$ corresponding to that time period will obtain a value close to 1. In a sense, we can say that our system exploits the fact that the vocabulary of tweets gets more specific when something important happens within an event, and therefore, the weight of the edges between the corresponding terms gets higher.

Let $\mathbf{c} = \mathbf{W}\mathbf{x}^*$. The closer the values of the entries of $\mathbf{c}$ to those of $\mathbf{b}$, the lower the value of the objective function of the optimization problem (1). As mentioned above, we are interested in detecting pairs of terms which co-occur in many posts of the current time period, and which appeared only at a limited number of posts in the previous periods. We assume that such pairs of terms are indicative of major developments of an event. These pairs of terms force the objective function of the optimization problem (1) to take large values. Our detection mechanism is based on that value. However, pairs of terms which co-occurred in many tweets of the previous time periods can also cause the objective function to take large values. To account for this, we set the entries of vector $\mathbf{c}$ that are larger than these of vector $\mathbf{b}$ equal to those of $\mathbf{b}$, that is

$$\mathbf{c}_i = \min(\mathbf{c}_i, \mathbf{b}_i), \quad \forall i \in 1, \ldots, n^2$$

We then proceed to compute the value of the function $\frac{1}{2}||\mathbf{c}-\mathbf{b}||_2^2$. The higher the value of the above formula, the higher the probability that an important moment occurred during the current time period. Hence, to decide if an interesting sub-event has occurred, we compare it against a specified threshold θ (which is learned automatically as described in Sect. 4).

3.4 Summarization

After detecting a sub-event, the summarization module of the proposed system generates a textual description for that sub-event. We propose an extractive summarization algorithm which selects a subset of tweets that contain the most significant concepts of the sub-event. Specifically, given a set of tweets corresponding to the summary of the sub-event, we define a monotone submodular function which rewards both the coverage and the diversity of the set. We then employ a well-known greedy algorithm for optimizing this function [8].

We assume that tweets that contain multiple "important" edges capture most of the details of the sub-event. Given the graph-of-words representation G_i of the tweets posted during a time period i, and a set of tweets $\mathcal{S} \subseteq \mathcal{D}_i$, we define a function f which takes as input a set of tweets (i. e. $\mathcal{S}$) and is equal to the sum of the weights of the edges of G_i that are "covered" by tweets belonging to that set. The function is thus equal to the sum of the weights of all the edges that link all pairs of terms found in the input set of tweets. By maximizing function f given a cardinality constraint, we can generate a summary which is very informative and at the same time diverse. Let $\mathcal{S}$ represent the summary of a sub-event. It is easy to show that when we add a new tweet into our summary, the value of f never decreases (hence, f is monotone nondecreasing), and that it satisfies the property of diminishing returns: given two sets of tweets $\mathcal{A}, \mathcal{B}$ with $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{D} \backslash d$, then it holds that $f(\mathcal{A} \cup d) - f(\mathcal{A}) \geq f(\mathcal{B} \cup d) - f(\mathcal{B})$. Hence, f is a monotone nondecreasing submodular function and we can use a greedy algorithm to compute an approximate solution guaranteed to be within $(1 - 1/e) \approx 0.63$ of the optimal solution [14].

4 Experiments and Evaluation

In this Section, we evaluate the proposed system in the task of sporting event summarization. We have also applied the system to other types of events, such as the terrorist attack that took place near the British Parliament in London on 22 March 2017. Due to space limitations, the empirical results for that event are presented in the supplementary material[2].

[2] http://www.db-net.aueb.gr/nikolentzos/files/ecir18suppl.pdf.

Table 1. Summary of the 20 matches from the 2014 and 2010 FIFA World Cups that were used in our experiments.

Match	Actual events	Collected tweets	Preprocessed tweets
ARG - BEL	7	313,803	108,250
ARG - GER	9	824,241	262,112
AUS - NED	12	96,834	25,997
AUS - ESP	9	86,843	13,608
BEL - KOR	7	99,192	32,053
CMR - BRA	11	148,298	35,085
FRA - GER	6	525,725	160,727
FRA - NGA	6	367,899	128,718
GER - ALG	8	712,525	276,227
GER - BRA	12	973,985	295,875
GER - GHA	8	285,804	77,449
GER - USA	7	256,445	86,040
GRE - CIV	10	113,402	51,101
HON - SUI	8	41,539	10,082
MEX - CRO	11	155,549	36,981
NED - CHI	8	95,108	25,819
NED - MEX	10	628,698	217,472
POR - GHA	10	272,389	91,110
GER - SRB	14	45,024	29,062
USA - SVN	12	85,675	53,292
Total	185	6,128,978	2,017,060

Dataset. We evaluated the proposed system on a dataset consisting of football matches from the 2010 and 2014 FIFA World Cups that were collected using the Twitter Streaming API. Specifically, the dataset contains 18 matches from the 2014 FIFA World Cup and 2 matches from the 2010 FIFA World Cup. Table 1 shows statistics of the matches used in our experiments. The Table illustrates both the total number of tweets collected for each match and their resulting number after applying the preprocessing steps described in Sect. 3.1. It also shows the number of sub-events (considered sub-events are described in next Section) that occurred in each match. Overall, the dataset contains 6,128,978 tweets with an average of 306,448 and 100,853 tweets per match before and after preprocessing. Furthermore, the matches contain 185 sub-events in total.

Key Sub-events and Ground-Truth Construction. Following Nichols et al. [16] and Meladianos et al. [11], we considered 8 major sub-event types: (1) goals, (2) own goals, (3) red cards, (4) yellow cards, (5) penalties, (6) match starts, (7) end of matches, and (8) half time breaks. Other types of sub-events such as missed attempts or offsides were discarded as they either depend on the subjective opinion of the person who wrote the summary or their impact in the match outcome is limited and are not broad-casted by Twitter users.

The actual sub-events for each match and their corresponding textual descriptions were collected from the official website of FIFA[3]. To annotate the collected matches, we employed the following approach: we first set the length of the considered time frames equal to 60 seconds. Then, for each time frame, we employed the summarization module of our system to extract a textual description (i. e. a pair of tweets) given the set of tweets posted in this frame. We then had two humans to manually annotate these descriptions. More specifically, tweets that describe the key sub-event types with an intervening period of at most a few minutes between their time frame and the actual sub-event were labeled as positive. Time frames whose extracted textual description contain other sub-events such as missed attempts and substitutions, which are not included in our key sub-event types, were not taken into consideration for the evaluation. In fact, such sub-events provide useful information regarding the match and we would like them to be included into the summary. The remaining time frames were labeled as negative.

Parameter Learning. Since the volume of tweets of the 20 considered football matches varies a lot, we found it necessary to use a different value of threshold θ for each match. Some preliminary experiments provided evidence of a relationship between the optimal value of θ and the number of tweets posted during an event. We followed a supervised learning approach where we split the set of events (i. e. football matches) into a training and a test set. Specifically, we randomly selected 3 matches from the set to serve as training examples, while the remaining 17 matches were placed into the test set. For the 3 matches of the training set, we performed an exhaustive grid search for identifying the optimal value of threshold trying all possible values of $\theta \in [1, 100]$ with step 0.1. The optimal value of threshold was the one that led to the highest f1-score. We observed that there is a strong correlation between the optimum threshold θ and the total number of tweets posted during a match. Therefore, we employed the following linear model comprising of three parameters

$$\hat{\theta} = w_1 x^2 + w_2 x + w_3$$

where x is the number of tweets posted during the match, and w_1, w_2, w_3 are the three parameters. We then computed the values of the parameters that minimize the least-squares-error with regard to the optimal thresholds of the training set. At test time, we used the above model to set the threshold θ of each match based on the total number of tweets related to that match.

Baseline. As mentioned above, most approaches in the literature use tweet rate to detect sub-events. Hence, we implemented a detection system based on post rate to serve as the baseline for our evaluation. Given a stream of tweets,

[3] www.fifa.com/worldcup/archive/brazil2014/matches/index.html.

Table 2. Micro- and macro-average precision, recall and f1-scores on the 17 matches of the test set.

Method	Metric					
	Macro-average			Micro-average		
	Precision	Recall	f1-score	Precision	Recall	f1-score
OptSumm	0.76	0.75	0.75	0.73	0.74	0.73
Burst	0.78	0.54	0.64	0.72	0.54	0.62

Table 3. Key sub-event types, their actual numbers and detected numbers over the 17 matches of the test set.

Event type	# actual events	# detected events
Goal	42	42
Own goal	2	2
Penalty	3	3
Red card	3	3
Yellow card	51	15
Match start	17	14
Match end	17	17
Half time	17	17

all tweets posted within a specific time frame are first preprocessed following exactly the same procedure as in the case of the proposed system. Subsequently, the tweeting rate of the current time frame is computed and if it exceeds a specific threshold, the system considers that a sub-event has occurred. The value of the threshold was computed separately for each match using a linear model similar to the one presented above. The parameters of the model were optimized on the same set of matches as in the case of the proposed approach.

Experimental Results. We first evaluate the proposed system (OptSumm) and the baseline approach (Burst) on the task of sub-event detection. For these results, we use the set of manually annotated tweets described above. We report performance using standard measures in information retrieval such as *precision*, *recall* and *f1-score*. Note that for both approaches, the threshold for each match is determined using the parameters learned on the three matches of the training set as described above. Table 2 illustrates the micro- and macro-average precision, recall and f1-scores of the two approaches over the 17 matches of the test set. The proposed system clearly outperforms the baseline. Specifically, OptSumm managed to detect sub-events that could not be detected by Burst, leading to better recall and f1-scores.

We next investigate the ability of the proposed system in detecting sub-events that correspond to different types of plays in the match. Table 3 illustrates

Table 4. Summary of the France vs Nigeria match generated automatically using the proposed system and manually by a journalist on behalf of FIFA.

Our summary	FIFA
Underdog Nigeria vs. European giants France. Going to be a great match!	The match kicks off
Nigeria awarded a free kick in a good position after Matuidi collides with Odemwingie. Nigeria looking decent on the break so far. #fra #nga	Matuidi (France) concedes a free-kick following a challenge on Odemwingie (Nigeria)
France #fra 0-0 Nigeria #nig - Nigeria with Eminike score, but ruled out for offside, good decision 18min	Emenike (Nigeria) is adjudged to be in an offside position
Pogbaaaaaa!!! excellent skill! made that entire move and ended it with a superb volley but keeper made a good save	Pogba (France) sees his effort hit the target
Half time: France 0-0 Nigeria. Goalless in braşília. tight game	The referee brings the first half to an end
54: Blaise Matuidi gets the first yellow card of the game after a nasty challenge	Matuidi (France) is booked by the referee
Nigeria the best team by far. That usually means France will scrape a lucky win	–
Omg! #Benzema so close to scoring, just 2? inches short. Still 0-0 (Nigeria-France) in a suddenly very exciting match!	Benzema (France) sees his effort hit the target
If the French don't score in this game, it would be a miracle for Nigeria. France has been inches away from about 3 goals at this point	–
Goal France! Who else, but the future, Paul Pogba, heading into an open net. Finally les blues score. 1-0, 80th min	Pogba (France) scores!!
Goool! France scores in the 91st min! Partial score, France 2-0 Nigeria. #worldcup goal count - 147	Yobo (Nigeria) scores an own goal!!
Full-time: #fra 2-0 #nga. France book their spot in the quarter-final while Nigeria crash out of the 2014 Fifa world cup	The final whistle sounds

the number of times that each key sub-event was detected compared to the total number of the key sub-events in the 17 matches. The proposed system successfully detected all goals, own goals, penalties and red cards of the 17 matches. This is not surprising since these correspond to primary sub-events which trigger the majority of user tweets. The system also detected all match ends and half times, and almost all match starts. However, the system failed to detect the yellow cards consistently and this may be due to the fact that yellow cards are not of significant impact for the outcome of a match.

As regards the generated summaries, Table 4 compares the summary generated by our system for the match between France and Nigeria with the one created by humans on behalf of FIFA. The proposed summarization module produces very informative and reasonable textual descriptions of the important moments. The quality of the generated summaries remains the same for the other matches of our dataset. We believe that a person can get a great idea of what happened during the match by reading the event summary.

5 Conclusion

In this paper, we presented a system capable of generating real-time summaries of events using only status updates from Twitter. The experiments that we conducted on sporting events showed that our system clearly outperforms the dominant approach on the sub-event detection task, and also generates very informative and readable summaries.

References

1. Akbari, M., Hu, X., Nie, L., Chua, T.S.: From tweets to wellness: wellness event detection from Twitter streams. In: AAAI, pp. 87–93 (2016)
2. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: real-world event identification on Twitter. ICWSM **11**, 438–441 (2011)
3. Castillo, C.: Big Crisis Data. Cambridge University Press, Cambridge (2016)
4. Chakrabarti, D., Punera, K.: Event summarization using tweets. In: ICWSM, pp. 66–73 (2011)
5. Chierichetti, F., Kleinberg, J., Kumar, R., Mahdian, M., Pandey, S.: Event detection via communication pattern analysis. In: ICWSM, pp. 51–60 (2014)
6. Java, A., Song, X., Finin, T., Tseng, B.: Why we Twitter: understanding microblogging usage and communities. In: SNA-KDD, pp. 56–65 (2007)
7. Letsios, M., Balalau, O.D., Danisch, M., Orsini, E., Sozio, M.: Finding heaviest k-subgraphs and events in social media. In: ICDM Workshops, pp. 113–120 (2016)
8. Lin, H., Bilmes, J.: A Class of submodular functions for document summarization. In: ACL, pp. 510–520 (2011)
9. Lin, J., Roegiest, A., Tan, L., McCreadie, R., Voorhees, E., Diaz, F.: Overview of the TREC 2016 real-time summarization track. In: TREC, vol. 16 (2016)
10. Mackie, S., McCreadie, R., Macdonald, C., Ounis, I.: Comparing algorithms for microblog summarisation. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) CLEF 2014. LNCS, vol. 8685, pp. 153–159. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11382-1_15
11. Meladianos, P., Nikolentzos, G., Rousseau, F., Stavrakas, Y., Vazirgiannis, M.: Degeneracy-based real-time sub-event detection in Twitter stream. In: ICWSM, pp. 248–257 (2015)
12. Meladianos, P., Tixier, A.J.P., Nikolentzos, G., Vazirgiannis, M.: Real-time keyword extraction from conversations. In: EACL, pp. 462–467 (2017)

13. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts. In: EMNLP, pp. 404–411 (2004)
14. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions I. Math. Program. **14**(1), 265–294 (1978)
15. Nenkova, A., McKeown, K.: A survey of text summarization techniques. In: Aggarwal, C., Zhai, C. (eds.) Mining Text Data, pp. 43–76. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-3223-4_3
16. Nichols, J., Mahmud, J., Drews, C.: Summarizing sporting events using Twitter. In: IUI, pp. 189–198 (2012)
17. Nikolentzos, G., Meladianos, P., Rousseau, F., Stavrakas, Y., Vazirgiannis, M.: Shortest-path graph kernels for document similarity. In: EMNLP, pp. 1891–1901 (2017)
18. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to Twitter. In: NAACL-HLT, pp. 181–189 (2010)
19. Sharifi, B., Hutton, M.A., Kalita, J.K.: Experiments in microblog summarization. In: SocialCom, pp. 49–56 (2010)
20. Shen, C., Liu, F., Weng, F., Li, T.: A participant-based approach for event summarization using Twitter streams. In: NAACL-HLT, pp. 1152–1162 (2013)
21. Srijith, P., Hepple, M., Bontcheva, K., Preotiuc-Pietro, D.: Sub-story detection in twitter with hierarchical dirichlet processes. Inf. Process. Manage. **53**, 989–1003 (2016)
22. Walther, M., Kaisser, M.: Geo-spatial event detection in the Twitter stream. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 356–367. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36973-5_30
23. Weng, J., Lee, B.S.: Event detection in Twitter. In: ICWSM, pp. 401–408 (2011)
24. Zhao, S., Zhong, L., Wickramasuriya, J., Vasudevan, V.: Human as real-time sensors of social and physical events: a case study of Twitter and sports games. arXiv:1106.4300 (2011)
25. Zhou, X., Chen, L.: Event detection over twitter social media streams. VLDB J. **23**(3), 381–400 (2014)
26. Zubiaga, A., Spina, D., Amigó, E., Gonzalo, J.: Towards real-time summarization of scheduled events from Twitter streams. In: HT, pp. 319–320 (2012)