Degeneracy-based Real-Time Sub-Event Detection in Twitter Stream

Polykarpos Meladianos
AUEB
pmeladianos@aueb.gr

Giannis Nikolentzos *
AUEB and IMIS / RC ATHENA
nikolentzos@aueb.gr

François Rousseau
Ecole Polytechnique
rousseau@lix.polytechnique.fr

Yannis Stavrakas †
IMIS / RC ATHENA
yannis@imis.athena-innovation.gr

Michalis Vazirgiannis
Ecole Polytechnique and AUEB
mvazirg@aueb.gr

Abstract

In this paper, we deal with the task of sub-event detection in evolving events using posts collected from the Twitter stream. By representing a sequence of successive tweets in a short time interval as a weighted graph-of-words, we are able to identify the key moments (sub-events) that compose an event using the concept of graph degeneracy. We then select a tweet to best describe each sub-event using a simple yet effective heuristic. We evaluated our approach using human-generated summaries containing the actual important sub-events within each event and compare it to two baseline approaches using several performance metrics such as DET curves and precision/recall performance. Extensive experiments on recent sporting event streams indicate that our approach outperforms the dominant sub-event detection methods and constructs a human-readable event summary by aggregating the most representative tweets of each sub-event.

1 Introduction

Humans are social by nature and they always seek opportunities for social interaction. This explains why social media have gained extreme popularity among Internet users in recent years. One of the most representative examples of social media is Twitter, a microblogging service that was launched in 2006 and that allows users to publish short messages, the so-called tweets, which are up to 140 characters long. Most people use the service to report latest news or to comment live events (Java et al. 2007). The messages posted by such kind of users tend to reflect a variety of events as they happen. The service facilitates the spread of news and allows users to discuss events as they occur. The events that are discussed by users vary both in type and in scale. Users may write posts about local events such as local festivals (Lee and Sumiya 2010), about political issues such as the upcoming elections (Tumasjan et al. 2010), or about more serious topics such as the protests that followed the Iranian elections in 2009 (Kwak et al. 2010). In some cases, news appear on Twitter faster than in any traditional news media. For example, it has been reported that disastrous events such as earthquakes can be detected in real-time by monitoring tweets (Sakaki, Okazaki, and Matsuo 2010). In addition to this, the opinions of people that publish messages about an event could provide perspectives different from the ones that are communicated by the traditional media.

Twitter can thus serve as a tool for the real-time identification of events, a rather challenging task for which several approaches have been proposed. However, events usually evolve (e.g., natural disasters, protests, etc.) and therefore, for most events, there is a set of sub-events nested within them. Users generate novel data when a new sub-event occurs. The content of the new data is different from the previous and it represents the current state of the event. Event detection methods are unable to detect new occurrences within an event as all the occurences share some common vocabulary and the corresponding tweet rates are relatively low. Hence, novel methods for sub-event detection have to be proposed.

Users can get information on a topic by using Twitter's search interface. However, this service is somewhat inefficient mainly for three reasons: (1) the large volume of returned tweets which is likely to overwhelm the user, (2) the redundancy of information between many messages that share the same textual content and (3) the noisy nature of Twitter – not only it has been infiltrated by large amount of spam but its 140-character limitation favors the use of abbreviations, irregular expressions and infrequent words leading to messages that are hard to read or interpret. This heterogeneity and daunting scale of the data poses a serious challenge to anyone who is interested in knowing more about a highly discussed topic. Summarization can serve as a solution to the problem of organizing and searching through Twitter's large corpus. Hence the need for an automated summarizer that can extract the main information regarding each sub-event and generate a summary that best describes the chain of events. Such an automated summarizer could provide a real-time overview of how these events are unfolding and could be of valuable help to users.

It has been reported that the volume of tweets reaches high levels around important moments (Marcus et al. 2011). Most

*Supported by the EU/Greece funded KRIPIS Action: MEDA Project with code 448842.

†Supported by the national "COOPERATION 2011" programme, project with code 11SYN 1 531 entitled "Informed Real-Estate Services: Leveraging Web 2.0".

sub-event detection systems rely on this claim to detect if an interesting occurrence within the main event took place. In this paper, we present an alternative sub-event detection algorithm that does not consider the volume of tweets to signal a key moment but rather the vocabulary of these messages. Our algorithm can be applied to every type of events. It is not necessary for the events to have a specific underlying structure. For our experiments, we considered events for which the start time and duration is known. Specifically, we used the tweets published during some matches of a football competition. The main contribution of our work can be summarized as follows:

A novel real-time sub-event detection mechanism based on graph-of-words and graph degeneracy that is able to more accurately detect important moments of an evolving event in a totally unsupervised and out-of-the-box manner.

The rest of this paper is organized as follows. Section 2 provides a review of the related work. Section 3 presents our event summarizer. Section 4 evaluates the proposed approach and compares it with existing methods. Finally, Section 5 concludes the work.

2 Related Work

In this section, we review the related work published in the areas of *event detection* and *event summarization* in Twitter stream. Since Twitter has restrictions regarding the redistribution of its data, the works mentioned in this section do not test their methods on a common dataset but rather their own.

2.1 Event Detection in Twitter Stream

Recently, there has been substantial research activity in the area of event identification in Twitter. Considerable research efforts have been devoted on the identification of events of high importance by monitoring the Twitter stream. Mathioudakis and Koudas (2010) described a system that detects an event when a set of keywords appear together at an unusual and high rate. Weng and Lee (2011) proposed an event identification system called EDCoW that applies wavelet analysis on word frequencies to obtain new features for each word. Subsequently, it filters away trivial words that have low signal auto-correlations and it identifies events by clustering the remaining words using a modularity-based graph partitioning technique. Petrovic *et al.* (2010) proposed an algorithm where each new document is compared to the previous ones using locality-sensitive hashing for scalability reasons. Becker *et al.* (2011) first clustered the input Twitter stream and then trained a classifier on manually annotated data using temporal, social, topical and Twitter-specific features to distinguish events from non-events. Cataldi *et al.* (2010) detected emerging topics in Twitter by creating a directed graph of the emerging terms and locating its strongly connected components. The emerging terms are identified by comparing the frequency of each term in a given time period with the previous ones. Marcus *et al.* (2011) developed a system called TwitInfo to visualize and summarize events on Twitter. The events are detected by identifying temporal peaks in tweet frequency and are presented to the user using a timeline-based display that highlights the peaks of high tweeting activity. Valkanas and Gunopoulos (2013) presented a system that clusters users according to their geographical location and then monitors the emotional state of each group of users – when there is a sudden change in a group's emotional state, the system signals an event.

2.2 Event Summarization in Twitter Stream

The systems that were previously described show evidence of the potential of Twitter for event detection. However, given a scheduled event, is it possible to identify its key moments using Twitter? The detection of the important moments of an event is a topic that has not yet received much attention by the research community. It usually consists of two stages: (1) the detection of the important moments or sub-events of an event and (2) the generation of a summary giving details about the sub-event. Chakrabarti and Punera (2011) developed one of the first sub-event detection systems. The authors used posts from Twitter in order to generate summaries of American football games. Their algorithm learns the underlying structure and vocabulary of a football game using a modified HMM. However, their algorithm is applicable only to recurring events as the HMM must be trained on similar events to reach high performance standards and it is not effective on previously unseen types of events. Nichols *et al.* (2012) focused on summarizing World Cup football matches, detecting a sub-event when the volume of status updates exceeds a threshold value. This value is computed offline from basic statistics of the set of all slopes for that match. The authors also presented an online approach where the threshold is computed using a sliding window. The summary consists of a number of sentences that are extracted from the phrase graph introduced by Sharifi *et al.* (2010b) using various weighting schemes. Zubiaga *et al.* (2012) explored the real-time summarization of Copa America football matches, considering that a sub-event occurred if the tweeting rate in a time frame was above 90% of the previous rates. To produce a summary, the system weights the terms using the Kullback-Leibler divergence and then the tweets based on the weights of their terms and chooses the tweet with the highest weight. Zhao *et al.* (2011) used the tweet rate to detect sub-events in American football games, using a sliding window that they divide into two sub-windows. If the fraction between the tweeting rates of the two sub-windows exceeds a threshold, a potential sub-event is detected. Afterwards, a lexicon-based recognition method is employed to label the sub-event as game-related or filter it out in case it is a random sub-event. Shen *et al.* (2013) experimented with NBA matches and a conference event, clustering the tweets based on the entity they referred to. The authors employ a mixture model approach for sub-event detection that they train using EM. They apply this sub-event detection approach to each cluster to identify the important sub-events associated with their corresponding entities. They then merge these sub-events to identify all the important moments of the event. To generate a summary, the authors use the TF-IDF-based approach proposed by Sharifi *et al.* (2010a) to extract a representative tweet for each important moment. Finally, Chierichetti *et al.* (2014) presented a system that detects important moments within

an event by using non-textual features of the tweeting pattern. The authors used tweets posted during the 2010 FIFA World Cup as their main dataset, and they trained a logistic regression classifier which uses only tweet and retweet rates as its features. The authors did not provide any methods for generating a summary of the detected moments.

3 Event Summarizer

In our work, we developed a system capable of generating real-time summaries of scheduled events using status updates collected from the Twitter stream – by real-time we mean that a sub-event is identified as it occurs. The proposed system takes as input a set of tweets from the stream and outputs a summary of the event composed of selected tweets if any sub-event has been detected. It is a 3-step process: (1) *feature extraction*, (2) *sub-event detection* and (3) *tweet selection* as illustrated in Figure 1 below.

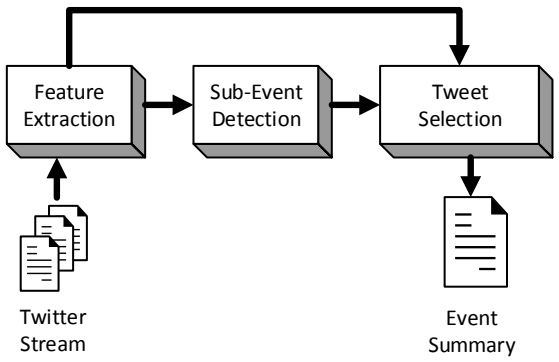

Figure 1: Overview of our proposed real-time sub-event detection and summarization system.

3.1 Feature Extraction

The first module extracts all the unique terms appearing in the set of tweets and assign them weights. These weights are used in the next two modules to decide if a sub-event has occurred and if so, what should be the summary produced for that sub-event. Therefore, the weighting scheme is of crucial importance to the system's functioning.

Graph-of-words We chose to represent the input set of tweets as a graph-of-words, following earlier approaches in keyword extraction and summarization (Erkan and Radev 2004; Mihalcea and Tarau 2004) and more recent ones in ad hoc IR (Blanco and Lioma 2012; Rousseau and Vazirgiannis 2013). We refer to the work of Blanco and Lioma (2012) for an in-depth review of the graph representation of texts.

The construction of the graph is preceded by a preprocessing phase where standard text processing tasks such as tokenization, stopword, punctuation and special character removal, and stemming are performed. Tweets with less than two tokens are also eliminated. Consequently, each remaining tweet consists of a multiset of unique terms and all the terms from all the tweets from the input set constitute the vertices of the graph-of-words. If two terms co-occur in any tweet of the input set, an edge is drawn between the two associated nodes. Therefore, a tweet can be seen as a fully-connected "subgraph-of-words". As opposed to previous works, the graph corresponds to a set of documents, not just one, and we do not use any sliding window of co-occurrence within a document to decide whether or not to add an edge, mainly because the size of a tweet is limited to 140 characters as opposed to Web pages (Blanco and Lioma 2012; Rousseau and Vazirgiannis 2013) for instance.

Regarding the edge weights, these are determined by the number of unique terms in each tweet. Consider the following tweet "*good goal by neymar*", then we want to represent it as a subgraph where each node has degree 1 (for reasons that will become clear in 3.1). Since it is fully connected, the degree of each unique term needs to be shared among all its adjacent edges that correspond to co-occurrences with the other unique terms of the tweet. If we consider that each co-occurrence is equally important, each edge of the subgraph should have a weight of $\frac{1}{n-1}$ where n is the number of unique terms in the tweet ($= 1/3$ in the example). The subgraph is then inserted in the graph-of-words, incrementing the edge weight by its weight in the subgraph if it already exists. For instance, assuming we then consider another tweet "*goal! neymar scores for brazil*", then the weight of the edge "goal–neymar" would be increased from $1/3$ to $1/3 + 1/4$.

Given the following additional two tweets "*watching the game tonight*" and "*goal !!! neymar scores again*", the resulting graph-of-words is illustrated on Figure 2 below.

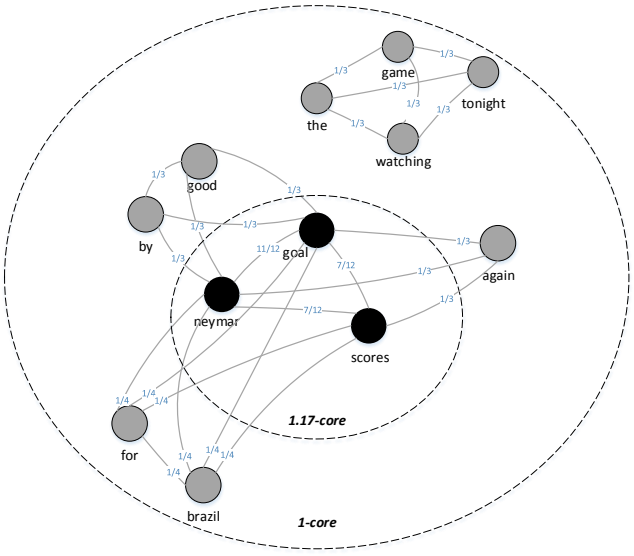

Figure 2: A graph-of-words built from 4 tweets and its k-core decomposition.

Once we have built a graph-of-words from the set of tweets, we can extract term weights using the graph degeneracy concept introduced next.

Graph degeneracy The idea of a *k-degenerate* graph comes from the work of Bollobs (1978, page 222) that was further extended by Seidman (1983) into the notion of a k-

core, which explains the use of *degeneracy* as an alternative denomination for k-core in the literature. Henceforth, we will be using the two terms interchangeably. Briefly, the k-core of a graph corresponds to the maximal connected subgraph whose vertices are at least of degree k within the subgraph. It naturally follows a decomposition of the graph into nested k-core of increasing cohesion (k), from the 0-core (the whole graph) to the k_{max}-core (the *main core*).

In weighted undirected graphs, the degree of a node is defined as the sum of the weights of its adjacent edges. Thanks to Batagelj and Zavernik (2002), the k-core decomposition of a weighted graph can be computed in linearithmic time ($\mathcal{O}(n + m \log n)$) and linear space ($\mathcal{O}(n)$) where n is the number of nodes and m the number of edges. The *core number* of a vertex is defined as the largest k for which the vertex belongs to the k-core. Intuitively, the core number is a more robust version of the node degree (Baur et al. 2007) and corresponds to how cohesive the node's neighborhood is. Figure 2 shows the k-core decomposition for the graph-of-words previously discussed.

Core number as term weight We extracted the k-cores from the weighted graph-of-words previously introduced and assign to each term the core number of its associated node as weight. We assumed that when a sub-event occurred in the time span corresponding to the input set of tweets, users would post messages containing information about that sub-event using the same set of terms, thus increasing the edge weights and the core numbers of these terms. Conversely, if nothing important happened, people would post "random" tweets without significant overlap in their vocabulary leading to low core numbers in the graph-of-words, ideally of 1. For example, in Figure 2, the terms of the "non-informative" tweet ("*watching second game of the evening*") belong to the lowest core while terms relevant to the sub-event ("*goal*", "*scores*", "*neymar*", "*brazil*") belong to a higher core.

We decided to consider each tweet ("subgraph-of-words") as a 1-core so that all tweets are equally treated independently of their number of terms. Had we set the weights of all the edges to a specific value, posts consisting of many terms would have been favored as they would belong to a higher core compared to posts with less terms. Hence, a node degree set to 1 as presented in 3.1 guarantees that every tweet is in the 1-core. That way, for a node to move from the 1-core to a deeper core in the overall graph-of-words, it needs to appear in multiple tweets with the same neighbors.

3.2 Sub-Event Detection

The second module aims at detecting sub-events based on either the tweeting rate like in related works or the "tweeting weight" like in our proposed approach.

Detection based on tweeting rate Previous research works (Shamma, Kennedy, and Churchill 2011; Marcus et al. 2011; Sankaranarayanan et al. 2009) showed that a sharp increase in the volume of status updates in Twitter indicates that something important (i.e. an *event*) has occurred. Several systems rely on this observation to identify emerging topics on Twitter in real-time. The effectiveness of these post

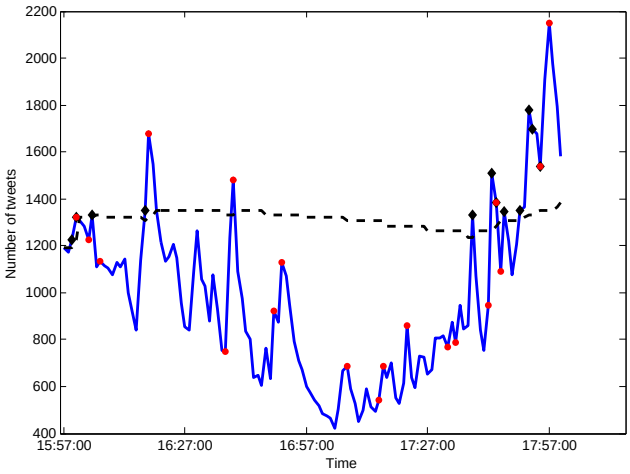

Figure 3: Tweeting rate for a football match, ground truth sub-events (red dots) and falsely detected sub-events (black diamonds and above dashed black line).

rate based methods in detecting events of high importance has been experimentally verified. Specifically, a wide variety of events ranging from celebrity deaths to plague outbreaks have been detected using this approach.

The above detection method has also been widely adopted for detecting important moments within an event. However, there are some subtle differences between the two types of event detection. First of all, in the case of event detection, our input is the whole Twitter stream and when a significant event occurs, due to the large number of users, the increase in the volume of tweets is clearly visible. Conversely, in the case of sub-event detection, the event that we are interested in is not as significant as the breaking news and its audience is limited only to people that are interested or participate in it. Therefore, the peaks in the histogram of tweeting rates are not always as high, hindering the task of sub-event detection. In addition, in the first case, the goal is to detect a standalone event that attracts user's attention, while in the second, there are many events in a short time span that must be detected.

To confirm our intuition, we investigated for a number of football matches if the ground truth sub-events all corresponded to large spikes in the volume of tweets. The result for the 2014 World Cup quarter-final match between Germany and France is shown in Figure 3. We notice that over the course of the match, the volume of status updates fluctuates. In general, the more important a sub-event is, the more users are likely to tweet about it almost immediately. For example, in a football match, there are some major sub-events such as goals and red cards where the tweeting rate skyrockets. Conversely, minor sub-events such as yellow cards and goalkeeper saves do not result in a high tweeting activity. Such a relatively low tweeting activity can also emerge from "random" tweets. It may happen as well that in a specific time frame people posted more tweets than usual and their increased activity was falsely detected as an important moment. The red dots in Figure 3 indicate the ground-truth sub-events and the black diamonds falsely detected sub-events

based on the tweeting rate. We see that they overlap as reported in related works but not completely, impacting both the precision and recall as we will see in the experiments.

Detection based on tweeting weight We assumed that if a sub-event occurred, users are likely to use terms from a specific vocabulary to describe it. For example, in the case of a save in a football match, messages in Twitter will contain with high frequency terms such as "save", "miss", "goalkeeper", the name of the goalkeeper and the name of the player that missed the attempt. We thus introduce a sub-event detection approach that relies on the frequency of the used terms. Instead of examining the volume of tweets, we examine the weights of the terms as determined by the weighting mechanism described earlier. If there is a sharp increase in them, we report the occurrence of a sub-event regardless of the number of tweets.

Each graph-of-words corresponds to a set of successive tweets within an interval of the stream. We set this interval to 60 seconds as in football matches, it is rather unlikely for two separate events to occur within the 60 seconds window and in case this happens, we observed that all the events can be described with a single tweet (e.g., foul followed by a red card). Thus, every 60 seconds our system builds a weighted graph-of-words using the tweets that were posted during that period. Figure 4 depicts a subgraph of the graph-of-the-words that was created when Germany scored in the 2014 FIFA World Cup final consisting only of the 8 terms belonging to the top 4 cores.

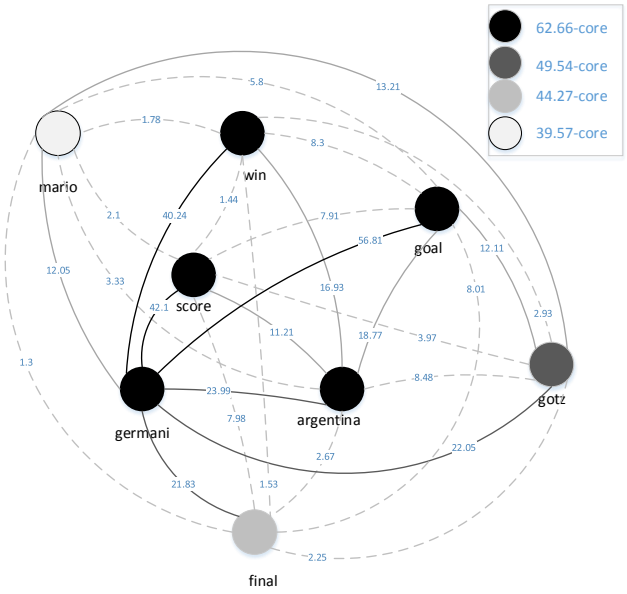

Figure 4: Four highest cores of the graph-of-words generated after Germany's goal in the 2014 FIFA World Cup final.

In preliminary experiments, we used an absolute threshold to detect important moments. Specifically, if the sum of the core numbers of the d terms belonging to the highest cores exceeded the threshold, the system signaled the occurrence of a sub-event. However, we observed that people often keep commenting on sub-events long after they have happened. Due to the use of common vocabulary, the core numbers of some terms corresponding to these sub-events remain high and we could detect the same sub-event multiple times. Hence, we decided to compare the sum of the core numbers of the d terms belonging to the highest cores with the average sum over the last p time periods. This way, the system estimates people's tweeting activity on a topic and only if their focused activity has increased, it signals the occurrence of a (new) sub-event.

More formally, the system considers that an important moment occurred when the sum of the core numbers of the d terms belonging to the highest cores for the current time period increases by at least a predefined threshold θ from the average sum over the last p time periods. Thus, if c_i^t is the core number of vertex i (sorted in descending core number) at time period t, we consider that an important moment occurred at a time period t if:

$$\sum_{i=1}^{d} c_i^t > \theta \times \frac{1}{p} \sum_{j=t-p}^{t-1} \sum_{i=1}^{d} c_i^j \quad (1)$$

The threshold can be interpreted as a parameter that measures how much more people comment on a specific topic compared to the previous time frames. For instance, for a threshold of 1.3 (θ), if people start using the same top 10 (d) terms of highest core number 40% more than in the past 10 (p) minutes then it probably means that a new sub-event occurred. Until 30%, either nothing is still happening or the same sub-event is still discussed.

We only take into account a fixed subset of the core numbers (the top d) to prevent the detection from being biased towards time intervals with high tweeting activity. More specifically, the sum of the core numbers depends on the number of vertices in the graph and also on the number of tweets that were posted in the time interval. The core number of each unique term in the graph is at least 1 and therefore, the more tweets that are posted, the greater the probability that new terms appear in the graph, increasing the total sum of core numbers (in the extreme case, imagine thousands of unique terms that would artificially increase the overall sum by 1 each). In our experiments we used $d = 10$, assuming that 10 unique terms were enough keywords to give an accurate and unique description of a sub-event. We also set the number of past periods considered p to 10 since for a football event specifically, some people continue discussing a sub-event for 10 minutes after its occurrence. Note that this does not prevent the system from detecting a new sub-event as its vocabulary will be different.

With regards to other sub-event detection methods that can be found in the literature, the outliers sub-event detection method proposed in (Zubiaga et al. 2012) would not be a good candidate for identifying interesting occurrences in our case. The method recognizes a sub-event if the tweeting rate is above at least 90% of the tweeting rates of the previous periods. Figure 3, as aforementioned, shows the tweeting rates of the 2014 FIFA World Cup quarter-final match between Germany and France along with the ground-truth sub-events. The dashed line represents the threshold of the

method used in (Zubiaga et al. 2012). The time frames with tweeting rate above the line are recognized as sub-events, while the ones under the line do not. As we can see, the outliers method would falsely detect a large number of time periods at the end of the match as sub-events while it would fail to detect several sub-events in the middle of the match.

3.3 Tweet Selection

The final component of the summarization system is only activated if a sub-event has been detected by the sub-event detection mechanism. Its task is to select the tweet that best describes the sub-event. From all the tweets that were posted in the previous 60 seconds, it extracts the tweet that contains the main information about the sub-event. For that purpose, we scored each tweet in the time period with the sum of its term weights (core numbers in the graph-of-words in our case). Indeed, if the weight of a term is high, the term is very likely to be related to the sub-event. In other words, the higher the weight of a term, the more representative the term of the interesting occurrence in the event. *We choose the tweet with the highest score to serve as the textual description of the sub-event.* Our tweet selection mechanism favors posts consisting of a large number of terms whose sum of weights is likely to be greater compared to posts consisting only of a few terms. However, the length of a message is already limited and we also observed empirically that large posts provide more accurate descriptions of the events.

4 Experiments and Evaluation

In this section, we first describe the dataset that we used for our experiments and the preprocessing steps we followed. We next give a description of the approaches against which we compared our method, as well as details about the annotation process. We last report on the performance of our algorithm and we provide arguments to why our approach can also detect other kinds of events.

4.1 Data Description

The dataset that was used to evaluate the effectiveness of our method contains tweets from several football matches that took place during the 2014 FIFA World Cup in Brazil, between the 12^{th} of June and the 13^{th} of July 2014 and were collected using the Twitter Streaming API[1]. Since our approach is used for sub-event detection, each match was considered as a standalone event. The dataset contains several millions of tweets with an average of $685,232$ tweets per match and an average tweet rate per minute of $5,162$ (the distribution is left-skewed since matches closer to the final have a much larger volume of tweets than the initial ones).

Although the initial dataset consisted of exactly 30 matches, we discarded 10 of them due to missing tweets for several time periods. Since, as we discuss later, the evaluation process is a rather daunting task, we randomly picked 11 out of the 20 remaining matches for our experiments. Table 1 shows statistics of the matches that were used for the evaluation. We can see that the volume of tweets of the 11 matches varies a lot, which makes the dataset ideal for testing various approaches, since the algorithms will have to be robust and run effectively on any type of matches, active or not in terms of tweeting activity. The matches contain a total of 95 sub-events considering the definition of section 4.4. Additionally, we tested our approach on two matches from the 2010 FIFA World Cup that are described in (Nichols, Mahmud, and Drews 2012), using the exact same dataset (the tweet ids were kindly provided by the author). For the third match that was evaluated in that paper, we did not possess a full set of tweets and was therefore excluded from the evaluation.

Match	# sub-events	# tweets
Germany vs. Argentina	8	$1,907,999$
Argentina vs. Belgium	7	$1,355,472$
France vs. Germany	6	$1,321,781$
Honduras vs. Switzerland	7	$168,519$
Greece vs. Ivory Coast	10	$251,420$
Croatia vs. Mexico	11	$600,776$
Cameroon vs. Brazil	11	$532,756$
Netherlands vs. Chile	7	$301,067$
Australia vs. Spain	9	$252,086$
Germany vs. Ghana	8	$718,709$
Australia vs. Netherlands	11	$126,971$
All Matches	95	$7,537,556$

Table 1: Summary of the eleven 2014 FIFA World Cup matches that were used in our experiments.

4.2 Data Preprocessing

All social media including Twitter are very sensitive to acts such as spamming, trolling and flaming that lead to very noisy datasets, which are then hard to process. In addition, the performance of the sub-event detection and summarization algorithms may be largely affected by such kind of posts. As a result, the preprocessing step is a crucial task and should be considered carefully. In this subsection, we give details about the preprocessing steps that we followed. Since, for our experiments, we fed the data to our system in a streaming fashion, these steps were applied to the set of tweets included in 60 seconds temporal intervals.

The first step of the preprocessing task is to remove all the retweets contained in the data. We removed them in an automated manner as it became obvious from early experiments that retweets increase (1) the delay of the event detection mechanism and (2) the quantity of noise since it is not rare for a retweet to go 'viral'; overshadowing the effect of all the other posts. In addition, we observed that some users instead of using the retweet mechanism provided by Twitter, copied the contents of a post creating a new duplicate tweet. Such posts have the same effects as retweets and were removed. Besides retweets, we also removed answers to specific users by removing any tweet that contains the character "@" as in most cases they are not relevant to the event under consideration. In addition, it was observed that tweets containing URLs usually do not contain much textual information as the main information is provided by the web source the URL

[1] https://dev.twitter.com/streaming/overview

links to and in many cases, such tweets turn out to be spam messages (Dong et al. 2010). Therefore, we decided to eliminate them as well. Note also that we considered only English tweets by filtering the stream using the language field provided by Twitter along with the `jlang`[2] library for language detection since the language information provided by Twitter is not totally reliable. The remaining posts constitute the input set of tweets we discussed in Section 3.

4.3 Considered Approaches

The event summarizer illustrated in Figure 1 consists of 3 components: (1) a feature extraction module responsible for the *term weighting* scheme; (2) a sub-event detection mechanism to identify the occurrence of a sub-event based on either the *tweeting rate* or the *tweeting weight*; and (3) a tweet selection part to elect the most representative tweet based on a *tweet score*.

In the literature (Zubiaga et al. 2012; Nichols, Mahmud, and Drews 2012; Zhao et al. 2011; Shen et al. 2013), the baseline usually considered a sub-event detection based on the tweeting rate (*Rate*) and a tweet score based on term frequencies (*Freq*). In our work, we propose a sub-event detection based on the tweeting weight (*Weight*) and a term weight/tweet score based on core numbers in a graph-of-words (*Core*). It results the four following approaches:

- *Rate–Freq*: the common baseline.
- *Weight–Core*: our approach described in Section 3.
- *Weight–Freq*: a baseline similar to our approach but that uses the raw term frequencies instead of the core numbers for sub-event detection and tweet selection.
- *Rate–Core*: an alternative approach that only uses the core numbers for tweet selection.

4.4 Ground-Truth Sub-Events

In this subsection we briefly describe the steps taken in order to annotate the matches used in our experiments. The actual sub-events for each match and their respective summaries were collected on ESPN FC[3], a football website that provided live coverage of the matches during the competition. For our experiments, we considered the same key event types as Nichols *et al.* (2012): *goals, disallowed goals, match starts and stops, red and yellow cards, half-time breaks and penalties*. Other events like missed attempts or fouls were not considered as they either depend on the subjective opinion of the person who wrote the summary or they were not described in a unique way that could automate the mining process from the ESPN FC website.

Our proposed approach and the three other approaches all output a set of tweets they consider a good summary of the match. The number of tweets that the four approaches output depends on a threshold. In order to evaluate the four approaches, we set this threshold to a value such that every time period of 60 seconds, a representative tweet was selected and added to the summary. We then had human editors manually annotate every tweet contained in these summaries. For matches that did not go to extra time, this corresponds to 130 tweets for each approach. These tweets were matched with the happenings in the match. Specifically, tweets that describe the key event types and were posted within a few minutes after the event took place were labeled as related to the event. Tweets describing other events common in football matches such as injuries, substitutions or missed attempts, which are not included in our eight key event types, were not taken into consideration for the evaluation of the approaches because they provide useful information about the match and in fact, we want them to be included in the summary. The remaining tweets were labeled as not useful. Although these tweets are related to the match under consideration, they do not provide any substantial information about it.

[2]https://github.com/melix/jlangdetect

[3]http://www.espnfc.com/fifa-world-cup/4/scores

4.5 Experimental Results

In this subsection, we present the obtained results on the 2010 and 2014 FIFA World Cup datasets using various evaluation metrics.

Evaluation Metrics Because this is mainly a detection task, we used the standard we used the standard metrics that are *precision*, *recall* and *F1-score*. We considered a sub-event as positive if it was detected by the approach, negative otherwise. Therefore, a true positive (*tp*) corresponds to a key event that was detected as a sub-event, a true negative (*tn*) to a non-key event that was not detected as a sub-event, a false positive (*fp*) to a non-key event that was detected as a sub-event and a false negative (*fn*) to a key event not detected as a sub-event. We ran the evaluation over all eleven games using both micro- and macro-averages. Macro-average F1-score corresponds to the arithmetic mean of F1-scores over the set of matches while micro-average F1-score to the harmonic mean of precision and recall computed over all the per-interval decisions. Statistical significance of improvement over the Rate–Freq baseline was assessed using the sign test for micro results and the Student's t-test for macro results ($p < 0.05$) (Yang and Liu 1999).

In addition, since both our algorithm and the baselines use a threshold to perform the detection, a single point of operation is not sufficient to describe the system's performance and in order to get a greater insight into the effectiveness of the four approaches, we plotted a Detection Error Tradeoff (DET) graph. The DET curve (Martin et al. 1997) is a ROC curve variant that plots the *missed detection probability* ($p_{miss} = {}^{fn}/_{(tp+fn)}$) versus the *false alarm probability* ($p_{fa} = {}^{fp}/_{(tn+fp)}$) for various system operating points. The system is considered to perform best at operating points that are closer to the lower-left of the graph (i.e. lower error probabilities). Regarding the overall performance, the area under the curve should be minimal.

Results for 2014 FIFA World Cup In Figure 5, we plot the micro-average DET curves comparing the proposed approaches to the baseline methods. It is clear that our approaches (Weight–Core and Rate–Core) outperform the baselines over the whole set of operating points. We

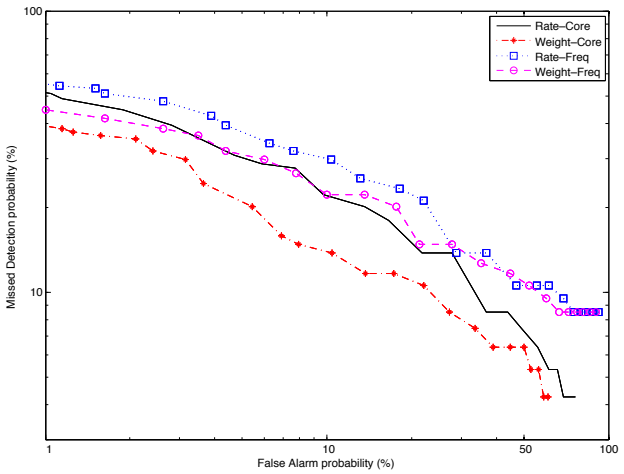

Figure 5: Micro-average DET curves over the eleven matches for all considered approaches.

Method	Micro-average F1-score	Macro-average (std) F1-score
Weight–Core	**0.68***	**0.72*** (0.12)
Rate–Core	0.61*	0.63 (0.20)
Weight–Freq	0.61	0.64 (0.20)
Rate–Freq	0.54	0.60 (0.25)

Table 2: Micro- and macro-average F1-scores over the eleven matches for all considered approaches.

also tried to find the threshold values for which each approach maximizes its performance and we discovered that all the four approaches perform best at threshold value 1.3, which corresponds to either 30% more tweets (Rate–* approaches) or 30% more cumulative weight (Weight–* approaches). Similar results were observed for macro-average DET curves but are not included for space constraints.

Our next step was to compare the four systems in terms of detection effectiveness at that optimal threshold. Table 2 illustrates the micro- and macro-average F1-scores of the four methods over the eleven matches. Bold font indicates the best results and * statistically significant improvement over the Rate–Freq baseline. Weight–Core managed to detect events that could not be detected by the other methods, especially at that scale ($7,500,000$ tweets) and to avoid the detection of periods without any occurrence of important moments, leading to better F1-scores. Note that Rate–Core and Weight–Freq yield to similar results, showing that it is the combination of tweeting weight and degeneracy-based term weighting that can achieve superior effectiveness.

Which key events are tweeted? To get an idea of which key event types are considered most important by Twitter users, we computed the number of times that each key event was detected by our system and compared it to the total number of the key events in the eleven matches. The results are shown in Table 3. Our algorithm can almost accurately identify goals, penalties, match ends and half times. It is rather

Event type	# actual events	# detected events
Goal	32	30
Penalty	2	2
Red Card	1	0
Yellow Card	27	14
Match Start	11	8
Match End	11	11
Half Time	11	10

Table 3: Key event types, their actual numbers and detected numbers by Weight–Core over the eleven matches. No disallowed goals were scored in any match.

surprising that our system did not manage to detect the one red card that was issued in the eleven matches. This red card was issued during Croatia vs. Mexico, which happens to be also the match that the two goals that our algorithm did not detect were scored. We decided to investigate this performance drop and we found that due to the fact that this match occurred at the same time as Cameroon vs. Brazil did, several tweets reported events of the later, adding noise to the dataset. Note also that the detection of match starts is not as accurate as the detection of half times and match ends. This may be due to the fact that our system started the process of the Twitter messages only five minutes before the start of the matches. Furthermore, 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. Thus, they are of lesser interest to the users in contrast to other event types such as goals for which users are very willing to update their statuses.

Results for 2010 FIFA World Cup Using the same threshold value of 1.3, the proposed system was further evaluated on the dataset containing tweets from two matches of the 2010 World Cup that were used in (Nichols, Mahmud, and Drews 2012) and the comparative results are shown in Table 4. It is clear that the degeneracy-based approach yields to better results in both matches. There is also an appreciable difference in precision and recall between these two matches and the eleven matches from the 2014 World Cup. Specifically, in these matches, both precision and recall are much higher than in the other matches. This is due to the fact that the tweeting rate histogram is much smoother for these matches compared to the previous ones as can be seen if one compares the tweeting rate curve from Figure 3 with the curve in (Nichols, Mahmud, and Drews 2012). These matches also contained less tweets than the 2014 matches because Twitter was perhaps not as popular in 2010 for commenting football matches. The interesting thing is that although the number of users and tweets have increased, the noise has also increased requiring much more sophisticated algorithms in order to successfully retrieve information. In the 2010 dataset, most of the key event types that we are interested in cause large increases in the tweet volume and are quite easy to detect, which seems to be no longer the case in 2014. This also shows something about the quality of the users. The fact that Twitter can now be considered as a "mainstream" social media has attracted a lot of users that

Match	# actual events	(Nichols, Mahmud, and Drews 2012)				Weight–Core			
		# detected events	Recall	Precision	F1-score	# detected events	Recall	Precision	F1-score
USA vs. Slovenia	13	8	0.62	**0.89**	0.73	13	**1.00**	0.87	**0.93**
Germany vs. Serbia	16	11	0.69	**0.92**	0.79	13	**0.81**	0.87	**0.84**

Table 4: Precision and recall for the two 2010 FIFA World Cup matches used in (Nichols, Mahmud, and Drews 2012).

Time	Our Summary	ESPN FC
8'	Goal!!!! Argentina!! After eight minutes Argentina lead Belgium by 1-0 scored by Higuain	Goal! Argentina 1, Belgium 0. Gonzalo Higuain (Argentina) right footed shot from the centre of the box to the bottom left corner.
45+2'	HT: Argentina 1-0 Belgium. Fantastic goal by Higuain gives Argentina the slight lead over the red devils.	First Half ends, Argentina 1, Belgium 0.
52'	52m - Belgium's Eden Hazard with the first yellow card of the game	Eden Hazard (Belgium) is shown the yellow card for a bad foul.
75'	Argentina 1 - 0 Belgium — Biglia booked a yellow card. Meanwhile, Chadli on for Eden Hazard.	Lucas Biglia (Argentina) is shown the yellow card for a bad foul.
90+5'	Well at least that goal makes them advance to the semi finals. Argentina gets the ticket to advance and Belgium goes home.	Match ends, Argentina 1, Belgium 0.

Table 5: Summary of the Argentina vs. Belgium match generated automatically using Weight–Core and manually by a journalist from ESPN FC.

for their own reasons misuse the hashtags and talk about irrelevant things within the topic of the matches, adding noise to the dataset, whereas in the past, the posts were more focused on the match itself and as a result, the spikes were actually indicative of the match's important moments, as can be seen from the comparison of the achieved precisions between the two datasets.

Generated summaries Regarding the summaries that are generated by our system, Table 5 compares some sample summaries generated by our system with the ones created by humans for the ESPN website. It is obvious that the simple method that we employed for summarization gives very informative and concise descriptions of the sub-events. Of course, this is true for the case of football matches, while more complex techniques may be required if the sub-event cannot be described by a single tweet which is at most 140 characters long. The time that is given in the table is the minute in the match in which the sub-event occurred and not the detection time. The system has an average delay of 90 seconds which drops significantly if one lowers the time interval parameter. The delay was measured using only sub-events that occurred during the first half of the match and assuming that the match started exactly at the time it was scheduled to start. For the second half, since the exact duration of the half time and the delays at the end of the first half are not precisely known, we cannot measure the delay with absolute certainty. In any case, the advantage of a sub-event detection system based on the social media is profound as it allows live tracking of events which is not feasible by using other sources where the information may be available hours after the sub-event took place.

4.6 Application to Other Types of Events

Our sub-event detection and summarization algorithms have been designed to generate summaries for any type of evolving event. Our system does not rely on any external knowledge about the event under consideration and it can be straightforwardly applied to other kinds of events. During the evaluation process, the system was not aware of the dataset's context. The only thing that it expects in the input is a stream of tweets. When dealing with other kinds of events, a parameter that should be set after careful consideration is the time interval. The value of this parameter depends on the event under consideration. In the case of football matches, given their small duration and continuous game-flow, we decided to use time periods of 60 seconds. For other not so rapidly evolving events, this parameter is likely to be set to higher values since small intervals would lead to unnecessary computations. A system could adapt by starting with a small interval and then dynamically increase it until a certain events-interval ratio is achieved and possible relations between the time interval and the tweet rate could also be investigated. Besides the 2010 and 2014 FIFA World Cups, we tested our system on a dataset containing tweets from the protests that took place in Turkey in June 2013. We do not have any ground-truth to evaluate our system's performance, but manually looking at the set of positives we observed that most of them provided substantial information about the event.

5 Conclusion

In this paper, we dealt with the problem of generating real-time summaries of events using only messages from Twitter as our source. We proposed an approach based on the concept of graph degeneracy applied on the graph-of-words that is constructed from the terms contained in the posts. Our algorithm exploits the fact that the vocabulary of tweets gets more specific when something important happens within an event as many people feel the need to post messages about it. The experiments that we conducted on football matches

from the 2010 and 2014 FIFA World Cups showed that our proposed approach clearly outperforms the baselines on the sub-event detection task, and also produces good summaries. In addition, our algorithm managed to detect the majority of the key sub-events during each match and it is our belief that a person can get a great idea of what happened during each event by solely reading the produced summaries.

Acknowledgements

The authors are indebted to Dr. Jeffrey Nichols for his interest and his willingness to share the ids of the tweets of the 2010 and 2014 FIFA World Cup matches.

References

Batagelj, V., and Zaveršnik, M. 2002. Generalized Cores. *The Computing Research Repository cs.DS/0202039*.

Baur, M.; Gaertler, M.; Görke, R.; Krug, M.; and Wagner, D. 2007. Generating Graphs with Predefined k-Core Structure. In *Proc. ECCS*.

Becker, H.; Naaman, M.; and Gravano, L. 2011. Beyond Trending Topics: Real-World Event Identification on Twitter. In *Proc. ICWSM*, volume 11, 438–441.

Blanco, R., and Lioma, C. 2012. Graph-based term weighting for information retrieval. *Information retrieval* 15(1):54–92.

Bollobás, B. 1978. *Extremal Graph Theory*. Academic Press, London.

Cataldi, M.; Di Caro, L.; and Schifanella, C. 2010. Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation. In *Proc. MDM/KDD*, 4:1–4:10.

Chakrabarti, D., and Punera, K. 2011. Event Summarization Using Tweets. In *Proc. ICWSM*, 66–73.

Chierichetti, F.; Kleinberg, J.; Kumar, R.; Mahdian, M.; and Pandey, S. 2014. Event Detection via Communication Pattern Analysis. In *Proc. ICWSM*, 51–60.

Dong, A.; Zhang, R.; Kolari, P.; Bai, J.; Diaz, F.; Chang, Y.; Zheng, Z.; and Zha, H. 2010. Time is of the Essence: Improving Recency Ranking Using Twitter Data. In *Proc. WWW*, 331–340.

Erkan, G., and Radev, D. R. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research* 22(1):457–479.

Java, A.; Song, X.; Finin, T.; and Tseng, B. 2007. Why We Twitter: An Analysis of a Microblogging Community. In *Proc. WebKDD/SNAKDD*, 56–65.

Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is Twitter, a Social Network or a News Media? In *Proc. WWW*, 591–600.

Lee, R., and Sumiya, K. 2010. Measuring Geographical Regularities of Crowd Behaviors for Twitter-based Geo-social Event Detection. In *Proc. GIS-LBSN*, 1–10.

Marcus, A.; Bernstein, M. S.; Badar, O.; Karger, D. R.; Madden, S.; and Miller, R. C. 2011. TwitInfo: Aggregating and Visualizing Microblogs for Event Exploration. In *Proc. CHI*, 227–236.

Martin, A.; Doddington, G.; Kamm, T.; Ordowski, M.; and Przybocki, M. 1997. The DET Curve in Assessment of Detection Task Performance. Technical report, DTIC Document.

Mathioudakis, M., and Koudas, N. 2010. TwitterMonitor: Trend Detection over the Twitter Stream. In *Proc. SIGMOD/PODS*, 1155–1158.

Mihalcea, R., and Tarau, P. 2004. TextRank: Bringing Order into Texts. In *Proc. EMNLP*, 404–411.

Nichols, J.; Mahmud, J.; and Drews, C. 2012. Summarizing Sporting Events Using Twitter. In *Proc. IUI*, 189–198.

Petrović, S.; Osborne, M.; and Lavrenko, V. 2010. Streaming First Story Detection with Application to Twitter. In *Proc. NAACL HLT*, 181–189.

Rousseau, F., and Vazirgiannis, M. 2013. Graph-of-word and TW-IDF: New Approach to Ad Hoc IR. In *Proc. CIKM*, 59–68.

Sakaki, T.; Okazaki, M.; and Matsuo, Y. 2010. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proc. WWW*, 851–860.

Sankaranarayanan, J.; Samet, H.; Teitler, B. E.; Lieberman, M. D.; and Sperling, J. 2009. TwitterStand: News in Tweets. In *Proc. GIS*, 42–51.

Seidman, S. B. 1983. Network Structure and Minimum Degree. *Social Networks* 5(3):269–287.

Shamma, D. A.; Kennedy, L.; and Churchill, E. F. 2011. Peaks and Persistence: Modeling the Shape of Microblog Conversations. In *Proc. CSCW*, 355–358.

Sharifi, B.; Hutton, M.-A.; and Kalita, J. K. 2010a. Experiments in Microblog Summarization. In *Proc. SocialCom*, 49–56.

Sharifi, B.; Hutton, M.-A.; and Kalita, J. 2010b. Summarizing Microblogs Automatically. In *Proc. NAACL HLT*, 685–688.

Shen, C.; Liu, F.; Weng, F.; and Li, T. 2013. A Participant-based Approach for Event Summarization Using Twitter Streams. In *Proc. NAACL HLT*, 1152–1162.

Tumasjan, A.; Sprenger, T. O.; Sandner, P. G.; and Welpe, I. M. 2010. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In *Proc. ICWSM*, 178–185.

Valkanas, G., and Gunopulos, D. 2013. How the Live Web Feels About Events. In *Proc. CIKM*, 639–648.

Weng, J., and Lee, B.-S. 2011. Event Detection in Twitter. In *Proc. ICWSM*, 401–408.

Yang, Y., and Liu, X. 1999. A re-examination of text categorization methods. In *Proc. SIGIR*, 42–49.

Zhao, S.; Zhong, L.; Wickramasuriya, J.; and Vasudevan, V. 2011. Human as Real-Time Sensors of Social and Physical Events: A Case Study of Twitter and Sports Games. *arXiv:1106.4300 [physics]*.

Zubiaga, A.; Spina, D.; Amigó, E.; and Gonzalo, J. 2012. Towards Real-time Summarization of Scheduled Events from Twitter Streams. In *Proc. HT*, 319–320.