StreamGrid: Summarization of Large Scale Events using Topic Modelling and Temporal Analysis

Emmanouil Schinas
[1]Dept. of Electrical & Computer Engineering
Aristotle University of Thessaloniki
[2]Information Technologies Institute
Centre for Research & Technology Hellas
manosetro@iti.gr

Symeon Papadopoulos
Information Technologies Institute
Centre for Research & Technology Hellas
Thessaloniki, Greece
papadop@iti.gr

Yiannis Kompatsiaris
Information Technologies Institute
Centre for Research & Technology Hellas
Thessaloniki, Greece
ikom@iti.gr

Pericles A. Mitkas
[1]Dept. of Electrical & Computer Engineering
Aristotle University of Thessaloniki
[2]Information Technologies Institute
Centre for Research & Technology Hellas
mitkas@eng.auth.gr

Abstract

Due to the increasing popularity of microblogging platforms, the amount of messages related to large scale public events reach impressive levels. Although such messages can be quite informative regarding different aspects of the main event, there is a lot of spam and redundancy that makes it challenging to extract insights regarding the event of interest. In this work we describe a summarization framework that captures the important moments of an event by using a combination of topic modelling and bursty activity detection. We propose a data structure named StreamGrid, that maintains the information of active topics in regular time intervals at several scales. This structure is used for the creation of concise summaries for any time interval. Finally, the evaluation on a large Twitter dataset around the Sundance Film Festival demonstrates the potential of the proposed framework.

1 Introduction

Due to their increasing popularity, micro-blogging platforms, and especially Twitter, have evolved into a powerful means for getting connected with real world events. In large scale public events, ranging from sport events, such as football matches, to political events and festivals, the users that are somehow involved in the event use social media to share their experiences and express their opinions. In many cases, these messages are quite informative and provide real-time coverage of the ongoing event and may be correlated with important variables related to the event, e.g. film ratings [13]. Thus, not surprisingly, the amount of event-related messages has reached impressive levels [1].

However, a significant percentage of micro-blogging messages can be considered as non-informative or spam. This fact combined with the huge number of messages, makes it very challenging for interested stakeholders, such as event organizers and enthusiasts, to monitor the evolution of the event and understand its important moments. In case of long-running events, this becomes even more difficult due to the existence of numerous sub-events occurring within the main event. Such sub-events have different durations and impact on the main event. In addition, a large portion of the messages contain conversations about other entities of interest associated with the event. In other words, an event-related stream of messages is quite diverse and

In: S. Papadopoulos, P. Cesar, D. A. Shamma, A. Kelliher, R. Jain (eds.): Proceedings of the SoMuS ICMR 2014 Workshop, Glasgow, Scotland, 01-04-2014, published at http://ceur-ws.org

noisy, with different associated topics, conversations among users, and spam messages. Thus, there is a profound need for event-based summarization methods that can produce concise multi-document summaries for any time interval of the event, covering its main aspects.

The framework we propose in this work aims to create topic-based summaries of large-scale events for arbitrary time durations by applying post-analysis on the stream of event related messages. First, we apply LDA topic modelling to discover the underlying aspects of the event. To support summarization, we create a 2D-array structure named StreamGrid. This maintains the information of each topic at each time interval. To create the grid we assign messages to the detected topics and divide topic-associated messages using regular time intervals. Next, we create timelines for the set of topics and use them to detect the set of active topics at each time interval by finding the bursty activity periods in them. A greedy algorithm is used to obtain a set of representative messages that maximize the coverage of the event by selecting the maximum possible number of active topics and minimize redundancy across messages at the same time. Finally, to demonstrate the potential of the proposed framework, we perform an experimental evaluation on a real-world dataset consisting of tweets around the Sundance Film Festival 2013.

The paper is organized as follows. Section 2 contains a brief survey of related methods and applications. Section 3 describes in detail the proposed framework. Section 4 presents an experimental case study on the Sundance 2013 dataset. We conclude the paper and describe future work in Section 5.

2 Related Work

A substantial body of work exists in literature on the problem of micro-blogging summarization. A notable method for multi-document summarization relies on the computation of centroids based on content. Namely, the summary of a set of documents, represented as $tf \cdot idf$ vectors, consists of those documents that are closest to the centroid of the set [12]. Sharifi et al. [15] propose a method for the generation of a single sentence from a set of tweets, by using a graph-based technique. Nichols et al. [11] describe an algorithm that generates a summary of sports events. They use a peak detection algorithm to detect important moments and then apply the method of [15] to extract summary sentences from the tweets around these moments. The work of [8] uses linear-programming optimization to select summary sentences from tweets related to trending topics. Notably, they also make use of linked Web content to extend the original sources of information.

Shen et al. [16] present a participant-based approach for event summarization. A mixture model is proposed to detect sub-events at participant level, and the $tf \cdot idf$ centroid approach is used to create a summary of each sub-event. Similarly, Chakrabarti and Punera [4] propose the use of a Hidden Markov Model to obtain a time-based segmentation of the stream that captures the underlying sub-events. Alonso and Shiells [2] create timelines for football games, annotated with the key aspects of the event. Dork et al. [5] propose an interface for large scale events that employs several visualizations for interactive presentation of the event.

A different problem is tackled by Wang et al. [19]. Unlike other methods, that method aims to create a storyline from a set of event-related objects. A multi-view graph of objects is constructed, where the two type of edges capture the contextual similarity and the temporal proximity among objects. Then a time-ordered sequence of important objects is obtained via graph optimization. Lin et al. [7] extends the previous work to generate storylines from a set of micro-blog messages for arbitrary queries. To achieve this, they use query expansion techniques to retrieve the query-related messages and then apply the same method as [19] to create the storyline.

Another approach for summarizing evolving tweet streams is proposed by the Sumblr framework [17]. This relies on an online clustering algorithm for tweets and on maintaining distilled statistics of the clusters at specific time snapshots using a structure, named Pyramidal Time Frame. Then, a summarization technique is employed for generating summaries of arbitrary time durations based on the LexRank method [6].

3 Proposed Method

An overview of the proposed method is illustrated in Figure 1. The proposed framework processes a stream of online messages around an event and extracts informative summaries for any requested time duration. In other words, the proposed framework identifies a set of topics and then selects related messages based on their importance.

3.1 Topic Modelling

Topic modelling is based on the assumption that each document can be described as a random mixture of topics and each topic as a multinomial distribution over terms. In our approach we employ topic modelling by using the well known Latent Dirichlet Allocation model [3] across the whole stream of messages. This process is applied after the end of the event, when all the messages are available. However, topic modelling in micro-blog messages is problematic due to the

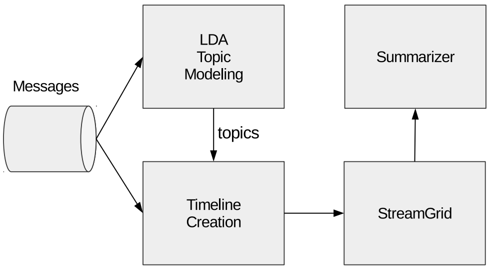

Figure 1: The StreamGrid framework

short length of their text. To overcome this, a lot of approaches have been proposed. To avoid changes on standard LDA, a relative simple solution is message pooling, in which messages are pooled together to form larger documents. We experimented with four methods of message pooling in a similar way as [10]. First, we tried to merge messages using constant length time bins. Then, we merged messages of the same author to form a single document. As a third option, we pooled messages together based on their hashtags. Messages with multiple hastags assigned to multiple documents and messages without any hashtag were assigned to the document with the highest textual similarity. As a fourth option, we used a 1NN clustering algorithm to cluster messages with high textual similarity. Each of those clusters formed a single document for the LDA method. In addition, for all of the pooling methods we filtered out messages having only one term and removed standard stopwords to discard the non informative terms.

Another drawback of LDA is that the number of topics must be defined; obviously, the number of topics in not known a priori in the context of large events. To determine the optimal number of topics for a given set of documents D we calculate two metrics, perplexity and average similarity across topics for different number of topics and choose a value that minimizes both metrics. For the calculation of perplexity we slit D into training and test documents, we estimate LDA over a range of possible numbers of topics using D_{train} and calculate the total perplexity of the documents in the test dataset D_{test} [18]. The perplexity of a document d given a trained model is defined as follows:

$$perplexity(d) = \exp \frac{-logP(d|\theta, \phi, \mathcal{G})}{L_d} \tag{1}$$

where L_d is the number of terms in document d, θ is the document-specific topic distribution, ϕ is the word distribution for topics, and $\mathcal{G}$ is the set of topics in the trained model. The total perplexity over dataset D_{test} is defined as

$$perplexity(D_{test}) = \exp \frac{\sum\limits_{d \in D} -logP(d|\theta, \phi, \mathcal{G})}{\sum\limits_{d \in D} L_d} \tag{2}$$

For the similarity between two topics, we calculate the Jaccard coefficient on the sets of top N terms of each topic.

3.2 StreamGrid Creation

After the detection of topics we have to associate messages with topics. We use the LDA model, estimated from the merged documents, to infer the probabilities of each message over the set of topics. We assign each message to the topic with the highest probability under the condition that this probability exceeds a predefined threshold. Although thresholding in this step leaves some messages unassigned, this is a desirable feature of the procedure as most of the unassigned messages are of low quality. In other words these messages can be considered as spam messages that cannot contribute any valuable information in the summary. Next, assignments are used for the creation of a data structure named StreamGrid. The first dimension of this grid comprises the detected topics and the second corresponds to time, divided into regular time intervals. Each cell $c(i, j)$ of StreamGrid contains the set of messages M_{ij} associated with $topic_i$, at time interval j. Each message m is represented as a $tf \cdot idf$ vector. The idf components are pre-computed over the whole set of messages. The tf part is the frequency of a term in the message normalized by the maximum frequency. Due to the short length of the documents in micro-blogging platforms, this component often equals to one. Using the set of associated messages in each cell, we calculate a merged $tf \cdot idf$ vector v_{ij}. In addition, we calculate a weight for each message and rank them according to it. The weight of a message m, associated with $topic_i$, in a specific time window j is defined as the sum of the weights of the terms contained in m. To calculate the weight of each term t, we use the following $tf \cdot idf$ scheme:

$$W(t, i, j) = tf_{ij}(t) \cdot idf(t) \tag{3}$$

$$W(m, j) = \sum_{t \in m} W(t, i, j) \tag{4}$$

where $tf_{ij}(t)$ is the frequency of term $t \in v_{ij}$ into the cell $c(i, j)$ of StreamGrid, and $idf(t)$ is the inverse document frequency over the whole corpus, $W(t, i, j)$ is the weight of term t in $c(i, j)$, and $W(m, j)$ the weight of message m in time interval j.

To detect the time intervals that a specific $topic_i$ of StreamGrid is active, we create a topic timeline by using time intervals as bins, and counting the associated messages of $topic_i$ in bin j. Then, we apply the peak detection algorithm used in [9] to detect time frames in the timeline that exhibit bursty behaviour. The algorithm identifies windows with high activity by finding significant increases in the timeline, compared to the historical mean value of activity. The time windows reported by the algorithm are used to set the active topics of each time interval. For example, if for a specific topic i, the algorithm identifies a time window $[a, b]$ with high activity, then we define all the time intervals $a \leq j \leq b$ as active moments of $topic_i$. After this step, the cells of StreamGrid, have a flag that indicates whether a specific cell is active or not. We use this flag to select a summary subset of messages, as described in the next paragraph. Also for each active $topic_i$ in a specific time interval j, we calculate a score that captures its significance over the rest of the active topics $\mathcal{A}$ in the same time interval.

$$Significance(topic_i, j) = \frac{|M_{ij}|}{\sum\limits_{topic_k \in \mathcal{A}} |M_{kj}|} \tag{5}$$

In addition, to have an overall estimation of the importance of each topic throughout the event, we calculate two measures for each topic using a similar approach as [14]. More specifically we define the peakiness of a topic as:

$$peakiness(topic_i) = \frac{max|M_{ij}|}{\sum\limits_{\forall j} |M_{ij}|} \tag{6}$$

and its persistence as

$$persistence(topic_i) = \frac{\underset{t_{peak}<j}{\text{avg}} \frac{|M_{ij}|}{\sum |M_{ij}|}}{\underset{j<t_{peak}}{\text{avg}} \frac{|M_{ij}|}{\sum |M_{ij}|}} \tag{7}$$

where t_{peak} is the time that the maximum peak of the timeline occurs.

3.3 Topic-Time Summarization

Our goal is to use the StreamGrid to summarize the event for an arbitrary time frame. As summary we denote a set of representative messages that mention the key aspects of the selected time period. Assuming that topics can capture these aspects, we use the active topics for that period to create a summary that meets the following criteria: a) as many aspects as possible are covered and b) redundancy due to near duplicate messages is minimized. To achieve this, we use an adapted version of the greedy algorithm used in [17]. The algorithm selects messages that are associated with different topics and that simultaneously have low degree of textual similarity between each other. The selection process is detailed by Algorithm 1. For an arbitrary time frame $\mathcal{F} = [a, b]$, we first find the sequence of time intervals in StreamGrid that covers $\mathcal{F}$. Then we get the set of active topics. A $topic_i$ is active in $\mathcal{F}$ if any cell $c(i, j)$ contained in $\mathcal{F}$ is active. Also, the significance score of an active topic in $\mathcal{F}$ is defined as the maximum significance score across all time intervals in $\mathcal{F}$. The weight $W(t, i, \mathcal{F})$ of a term t for $topic_i$ in $\mathcal{F}$ is defined as the sum of the weights in each cell $c(i, j) \in \mathcal{F}$. In a similar way, we define the weight $W(m, \mathcal{F})$ of message m over $\mathcal{F}$. Note that although a message belongs to a specific time interval, we use the term weights across the whole time frame to calculate the weight of m.

Algorithm 1 Topic-Time summarization

Input: StreamGrid, a time frame $\mathcal{F}$, length of summary L

Output: a summary set S

```
 1: S = $\emptyset$
 2: $\mathcal{A}$ = {set of active topics in $\mathcal{F}$}
 3: $M_c = \left\{ m | \underset{m}{\text{argmax}} W(m, i, \mathcal{F}), \forall i \in \mathcal{A} \right\}$
 4: while $|S| < L$ or $M_c \neq \emptyset$ do
 5:     for each message $m$ in $M_c$ do
 6:         calculate $score(m)$ according to Equation 8
 7:     end for
 8:     Select $m_{max} = \underset{m_i}{\text{argmax}}[score(m)]$
 9:     $S = S \cup \{m_{max}\}$
10:     $M_c = M_c - \{m_{max}\}$
11: end while
12: if $|S| < L$ then
13:     $M = \cup M_{ij}, \forall i \in \mathcal{A}, j \in \mathcal{F}$
14:     $M' = M - S$
15:     while $|S| < L$ do
16:         for each message $m$ in $M'$ do
17:             calculate $score(m)$ according to Equation 8
18:         end for
19:         Select $m_{max} = \underset{m_i}{\text{argmax}}[score(m)]$
20:         $S = S \cup \{m_{max}\}$
21:         $M' = M' - \{m_{max}\}$
22:     end while
23: end if
```

To produce a summary S of length L, the algorithm first gets the set of active topics as described above. Then, it collects the messages M_c with the highest weight $W(m, \mathcal{F})$ in each active topic (line 3). Through the lines 4-11, the algorithm, following a greedy approach, selects the messages that maximize the score of Equation 8. This consists of two parts weighted by a parameter a. The first part, measures the importance of the message, while the second the redundancy compared to the set of already selected messages. The importance of a message $m \in topic_i$ is a combination of two factors: a) the significance of the topic it belongs to, at this time frame, and b) the contribution of its textual content. To measure the redundancy of a message, we compute its average cosine similarity to the already selected messages. If the summary length is not reached, we perform the same selection process on the set of tweets that belong to the active topics (Lines 12-23).

$$score(m) = a*Importance(m) - (1-a)*Redundancy(m) \quad (8)$$

$$Importance(m) = Significance(i, \mathcal{F}) * W(m, \mathcal{F}) \quad (9)$$

$$Redundance(m, S) = \operatorname*{avg}_{m' \in S} Similarity(m, m') \quad (10)$$

4 Experiments

4.1 Dataset and event description

We conducted an evaluation of the proposed method on a dataset around the Sundance 2013 Film Festival that took place between January 15th and 30th, 2013. We used the Streaming API of Twitter to acquire tweets containing terms related to Sundance and posted during the event. More precisely, we collected all tweets containing the hashtags, #sundance, #sundance2013 and #sundancefest, and all the tweets that mentioned the official account of Sundance Film Festival (@sundancefest). This resulted in a dataset of 201,752 tweets. Among them, 100,046 were original tweets, while the rest of them were retweets. Although using three hashtags and one mentioned account covers only a subset of all possible tweets about the event, we consider this subset sufficiently representative as the vast majority of Twitter's users tend to adopt the official hashtags provided by organizers during events.

4.2 Topic detection

Figure 2 shows the perplexity and average similarity for different numbers of topics K. Although there is significant variance for the different values of K, the main trend for perplexity is to decrease as K increases. As we can see from Figure 2, the average similarity between all pairs of topics appears to stabilize for values of K larger than 100 topics. However, having a large number of topics creates topics with very few associated messages. We found that for $K > 200$ there is a substantial proportion of topics that have no associated message. Taking into account these facts, we set $K = 200$ for the rest of the evaluation. Regarding the pooling scheme, merging tweets having the same hashtags into single documents gave us the best performance with respect to perplexity and average topic similarity.

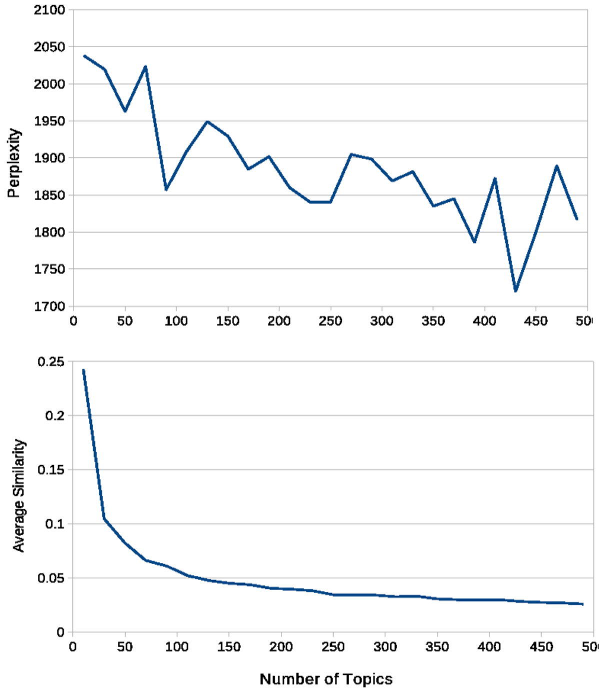

Figure 2: Perplexity and Average Similarity between topics for different number of topics K

4.3 StreamGrid Construction

The first part of Table 2 contains the top five topics with respect to the peakiness and the second one the topics with the highest persistence ratio. Examining the set of persistent topics we conclude that they can be divided into two main categories: The first comprises the truly persistent topics that are regularly discussed during the event, while the second category is made up of *multiplexed* topics that LDA failed to split further. This is due to the fact that some topics are conceptually different but share a similar set of related terms. This obviously affects summarization performance, as for each topic we select only the top weighted message. Thus, if the topic contains more than one concepts then the summarization algorithm selects only one concept and ignores the rest.

Figure 3 depicts the timelines of the same two sets of topics respectively. It becomes obvious that peaky

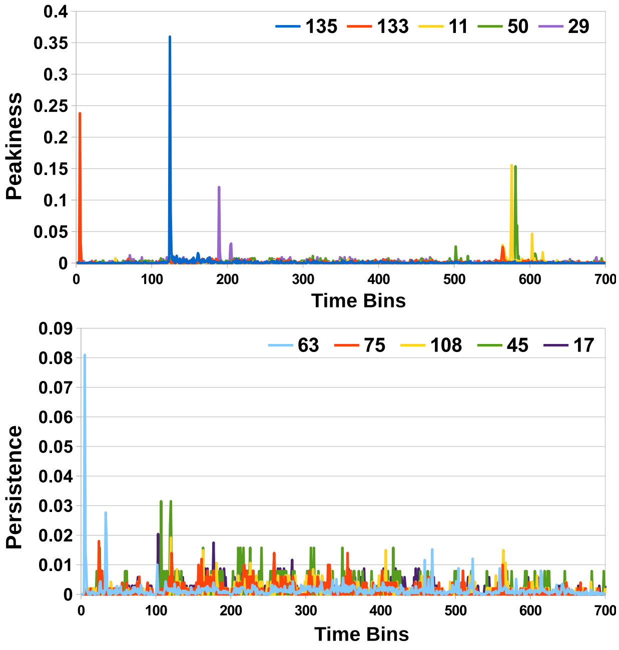

Figure 3: Timelines of the top five peaky and persistent topics

topics are highly localized, while persistent topics sustain for the whole duration of the event. To provide a visual representation of the StreamGrid structure over the whole duration of the event, we represent it as a heat map (Figure 4). The coloured cells in the grid represent the time intervals, in which the corresponding topics are active, and the color of the cell gives the significance of each active topic at this point. As shown in Figure 4, StreamGrid appears to be sparse, as only a few cells in it contain active topics. However, one can also observe several topics (rows) that exhibit consistent activity over the whole duration of the event.

4.4 Summarization

Baselines: To evaluate the summaries produced with StreamGrid, we used five baseline methods. Given an arbitrary time interval, we first get the set of messages posted during this interval and then we apply the following baselines to produce a summary of constant length L.

- *Random Summarizer:* For the set of tweets we choose randomly a subset of L tweets.
- *Popularity Summarizer:* We select the L most retweeted messages to form a summary. This favours the tweets that have attracted the attention of the audience. However, niche topics and potentially interesting events that gathered less attention tend to be missed.
- *$tf \cdot idf$ Summarizer:* We use the $tf \cdot idf$ weighting scheme described in the previous section to get the L highest weighted tweets.
- *Cluster-based Summarizer:* Instead of active topics, we divide the tweets of the time interval into L clusters using k-means clustering. For each cluster produced this way, we pick the highest weighted tweet using the $tf \cdot idf$ scheme.
- *LexRank Summarizer:* We create a graph where nodes represent tweets and the weights of edges between nodes represent their pairwise cosine similarity. The total weight of a tweet is the sum of the weights of the adjacent edges. The summary consists of the L highest weighted tweets in the graph.

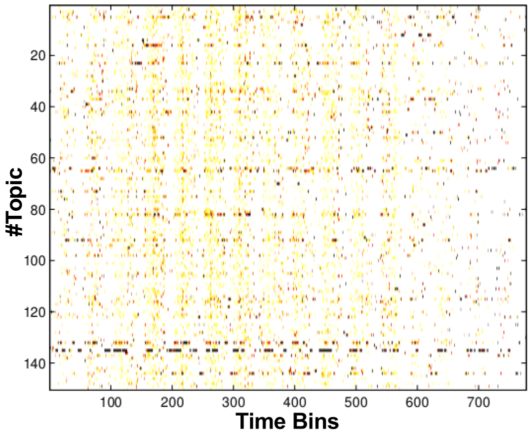

Figure 4: StreamGrid: Each cell of StreamGrid corresponds to a specific time interval and topic

Table 1: Details of five time intervals with the highest activity during Sundance Film Festival 2013

Start	End	#Tweets
Thu Jan 17 23:00	Fri Jan 18 00:00	1545
Sat Jan 19 19:00	Sat Jan 19 20:00	1477
Mon Jan 21 19:00	Mon Jan 21 20:00	1247
Sun Jan 27 03:00	Sun Jan 27 08:00	3735
Wed Jan 23 18:00	Wed Jan 23 21:00	1910

Finally, we compare the results of the StreamGrid Summarizer to the ones of the baseline methods for five time intervals that are connected with high activity during the main event. We detect these intervals by applying the peak detection algorithm of the previous section to the timeline of the whole dataset. We rank

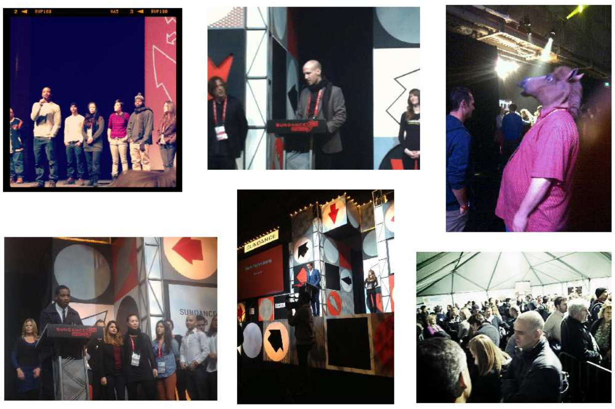

Figure 5: StreamGrid-based Multimedia Summary during awards ceremony (4^{th} row in Table 1)

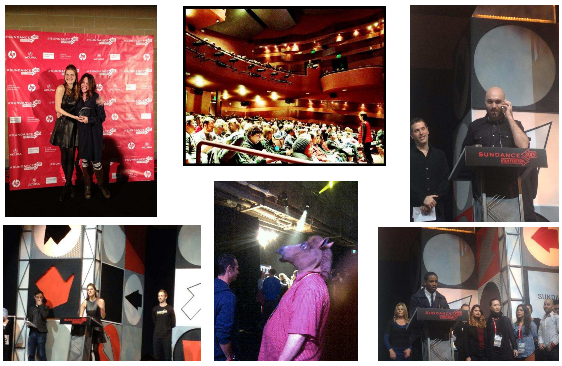

Figure 6: Multimedia Summary using most retweeted images during awards ceremony (4^{th} row in Table 1)

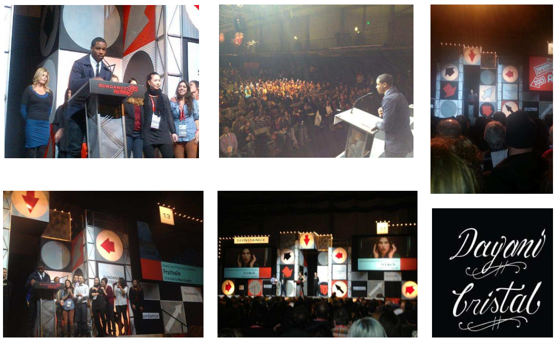

Figure 7: Multimedia Summary using LexRank during awards ceremony (4^{th} row in Table 1)

the detected bursts according to the rate of tweets and use the top five of them. The details of these intervals are provided in Table 1.

Table 3 contains summaries consisting of five tweets using StreamGrid and three of the baselines for the time period around the Awards Ceremony of Sundance Film Festival. Unsurprisingly, this is the time period with the highest peak during the event. During this period what may be reasonably considered as important pertains to the films that won awards. Such messages are usually posted by authoritative users and become highly retweeted. For this reason, summaries based on the number of retweets cover quite effectively the winning films. However, in other cases choosing very popular tweets does not lead to informative summaries. For example in the third time interval, the summary consists of tweets like "So freaking cool. #sundance http://t.co/C7a8rSaw" and "#Sundance day 4- leavin for Vegas now. Bye for now http://t.co/C2aRZnEC". These tweets were retweeted a lot, but may be considered as non-informative for the event. On the other hand, StreamGrid-based summaries for the Awards Ceremony contain tweets about winning films, even though these messages are not very popular. That is an indication that StreamGrid may detect an important topic even in cases that this does not attract attention from many users. Regarding the Cluster-based Summarization, an interesting feature is that avoidance of redundancy is inherent in the method, as similar messages are clustered together, and only the most weighted of them are selected for the summary. However, the weakness of the method is that not all clusters represent important aspects of the event.

Another indication of how topic modelling can improve summarization is the fact that StreamGrid, compared to the other baselines, tends to include tweets that mention films. The reason that this happens is that most of the topics detected by LDA are about films, so when the proposed summarization algorithm selects a set of tweets from the pool of active moments, this leads to the selection of film-associated tweets. We expect that, for other types of events, it will naturally generalize to other pertinent entities of interest that occur frequently, thus leading to the creation of topics. A noticeable disadvantage of baselines such as $tf \cdot idf$ and LexRank is the remarkable existence of redundancy. For example in case of LexRank four out of five tweets are related to the 'Fruitvale' film. This indicates that redundancy minimization is a necessary component of any summarization approach.

Finally, to evaluate how well the proposed method can create visual summaries, we apply it on the subset of tweets with embedded pictures. These tweets that comprise about 10% of the dataset create a consider-

Table 2: Examples of peaky and persistent topics

Peaky Topics			
Topic	**Representative Terms**	**Peakiness**	**#tweets**
135	paris, hilton, Blackfish, cnn, films	0.358	695
133	death, drink, countryman, sundance, charlie	0.247	588
11	lovelace, amanda, seyfried, portraits, premiere	0.161	1293
50	defeat, inevitable, pete, mister, film	0.143	267
29	butch, dynamite, android, worth, apps	0.123	323

Persistent Topics			
Topic	**Representative Terms**	**Persistence**	**#tweets**
63	hemingway, running, follow, crazy , marshall	3.963	2494
75	jehane, square, girlrising, premiere, screening	2.650	500
108	vhs, sequel , horror, review , time	2.318	469
45	afar, week, enjoy, ways, kicked	1.612	127
17	lindsay, lohan, canyons, blame, snubbed	1.557	343

ably sparser StreamGrid as the bursty periods in this subset are much fewer. An example of a multimedia summary using StreamGrid for the Awards Ceremony is shown in Figure 5. Comparing the StreamGrid-based multimedia summaries with the ones produced by the popular images (6), we observe that StreamGrid does not perform noticeably better in this task. This can be explained by the fact that tweets with embedded media have text of very low length and informativeness, which leads LDA to inferior performance with respect to the creation of representative topics and the assignment of messages to them. Regarding the redundancy in multimedia summaries, we found that using cosine similarity on the text of images as a metric of similarity between them is not appropriate to minimize redundancy. This can be seen in the LexRank-based summary in Figure 7. To this end, a combination of visual and textual features is foreseen as a more suitable means for discarding similar images.

5 Conclusion and future work

In this work, we proposed a framework for the summarization of micro-blogging messages during large scale events. The framework makes use of topic modelling to detect the underlying aspects of an event to the set of related messages. Then, for each topic it derives its temporal representation by associating messages to the discovered topics. Subsequently, a burst detection algorithm is used to find the important intervals for each topic. Finally, a greedy summarization algorithm generates summaries for arbitrary time intervals using the set of active topics for the same time duration. The results of experiments in a Twitter dataset around the Sundance Film Festival appear promising, demonstrating the potential of topic modelling on the multi-document summarization problem.

For future work, we first plan to compare our approach with competing summarization algorithms in a more systematic way, over more events and with the help of independent evaluators, with the goal of better capturing the subjective quality aspects of summarization. Taking into account the large number of topic modelling techniques that appeared in literature over the last years, we plan to investigate how the underlying model affects the summarization process. Furthermore, we intend to create a real-time version of StreamGrid, which could be used to get summaries of evolving and continuous streams of messages. To this end, we plan to employ more advanced topic modelling methods that can detect topic drift and unseen topics on new incoming messages. Finally, we will investigate methods to integrate popularity and user authority into the summarization process.

Acknowledgements: This work is supported by the SocialSensor FP7 project, partially funded by the EC under contract number 287975.

References

[1] Celebrating #SB48 on Twitter. `https://blog.twitter.com/2014/celebrating-sb48-on-twitter`, 2014. [Online; accessed 27-Feb-2014].

[2] O. Alonso and K. Shiells. Timelines as summaries of popular scheduled events. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 1037–1044. International World Wide Web Conferences Steering Committee, 2013.

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

Table 3: Summaries during awards ceremony (4^{th} line in Table 1)

Method	**Examples**
$tf \cdot idf$	Profound comment from @JoKiefer : Looper storyline echoes war on terror. Kill the terrorist before he becomes one? #Sundance13 #dirtywars #Sundance Institute Mahindra Global Filmmaking Award winnners include UK co-prodcution: Eva Weber: Let the Northern Lights Erase your Name #Sundance Institute Mahindra Global Filmmaking Award winnners include UK co-prodcution: Eva Weber: Let the Northern Lights Erase your Name #PussyRiot - A Punk Prayer takes home a World Cinema Doc Special Jury Award, directors Mike Lerner & Maxim Pozdorovkin: http://t.co/VmOP3tmg Gideon's Army was really eye-opening. I had no idea how many brave men and women have died trying to put Bibles in hotel rooms. #Sundance
LexRank	Yes! Audience Award U.S. Dramatic to 'Fruitvale' starring Wallace, aka Michael B. Jordan. Predict Grand Jury Prize, too #sundance Fruitvale wins audience award for U.S. dramatic; looks like it cld well be a sweep, w Grand Jury prize too. #Sundance @vulture: Yes! Audience Award U.S. Dramatic to 'Fruitvale' starring Wallace, aka Michael B. Jordan. Predict Grand Jury Prize, too #sundance @vulture: Start the Oscar watch now. 'Blood Brother' wins both Grand Jury and Audience Award for U.S. Documentary #sundance it's coming FRUITVALE wins the #Sundance Grand Jury Prize AND the Audience Award. Could not be happier. Congrats @fruitvalemovie and Ryan Coogler!
Popularity	#PussyRiot - A Punk Prayer takes home a World Cinema Doc Special Jury Award, directors Mike Lerner Maxim Pozdorovkin: http://t.co/VmOP3tmg Sebastian Silva's "Crystal Fairy" Wins #Sundance World Cinema Dramatic Directing Award: http://t.co/sKNn1Dqf Ryan Coogler's "Fruitvale" Wins #Sundance U.S. Dramatic Audience Award, presented by @AcuraInsider: http://t.co/Lknocyos JUST IN: Pinoy film "Metro Manila" wins the World Cinema Dramatic Audience award at the Sundance Film Festival — via @goldenglobes "The Spectacular Now" Wins #Sundance U. S. Dramatic Special Jury Award for actors Miles Teller Shailene Woodley: http://t.co/1Ouz2B7a
StreamGrid	Sebastian Silva's recorded speech: singing Hava Nagila while warping his face in Photo Booth. Word. #Sundance My pics of "The Spectacular Now" #Sundance Q&A , winner Special Jury award for acting to Miles Teller & Shailene Woodley http://t.co/G7r5QK1p #sundance Awards ... US Documentary ... Best Cinematography goes to Dirty Wars You know it would be hilarious if AUSTENLAND won the big #Sundance prize. Admit it, you'd soak that up. YES! "@indiewire: Ryan Coogler wins the audience award for FRUITVALE. #sundance http://t.co/NiVIzcTU" Grand Jury Prizes #Sundance: "Fruitvale" (dramatic) & "Blood Brother" (doc) FilmLinc list of winners: http://t.co/iyoeHuGz

[4] D. Chakrabarti and K. Punera. Event summarization using tweets. In *ICWSM*, 2011.

[5] M. Dork, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1129–1138, 2010.

[6] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, Dec. 2004.

[7] C. Lin, C. Lin, J. Li, D. Wang, Y. Chen, and T. Li. Generating event storylines from microblogs. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 175–184, New York, NY, USA, 2012. ACM.

[8] F. Liu, Y. Liu, and F. Weng. Why is "sxsw" trending? exploring multiple text sources for twitter topic summarization. In *Proceedings of the ACL Workshop on Language in Social Media (LSM)*, pages 66–75, 2011.

[9] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: Aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 227–236, New York, NY, USA, 2011. ACM.

[10] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 889–892, New York, NY, USA, 2013. ACM.

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

[12] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, 40(6):919–938, Nov. 2004.

[13] E. Schinas, S. Papadopoulos, S. Diplaris, Y. Kompatsiaris, Y. Mass, J. Herzig, and L. Boudakidis. Eventsense: Capturing the pulse of large-scale events by mining social media streams. In *Proceedings of the 17th Panhellenic Conference on Informatics*, PCI '13, pages 17–24, New York, NY, USA, 2013. ACM.

[14] D. A. Shamma, L. Kennedy, and E. F. Churchill. Peaks and persistence: Modeling the shape of microblog conversations. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 355–358, New York, NY, USA, 2011. ACM.

[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*, HLT '10, pages 685–688, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[16] C. Shen, F. Liu, F. Weng, and T. Li. A participant-based approach for event summarization using twitter streams. In *Proceedings of NAACL-HLT*, pages 1152–1162, 2013.

[17] L. Shou, Z. Wang, K. Chen, and G. Chen. Sumblr: Continuous summarization of evolving tweet streams. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 533–542, New York, NY, USA, 2013. ACM.

[18] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In L. Bottou and M. Littman, editors, *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 1105–1112, Montreal, June 2009. Omnipress.

[19] D. Wang, T. Li, and M. Ogihara. Generating pictorial storylines via minimum-weight connected dominating set approximation in multi-view graphs. In *AAAI'12*, pages –1–1, 2012.