

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/284104416>

MGraph: multimodal event summarization in social media using topic models and graph-based ranking

Article in *International Journal of Multimedia Information Retrieval* · March 2016

DOI: 10.1007/s13735-015-0089-9

CITATIONS

11

READS

484

4 authors:



Manos Schinas

Information Technologies Institute (ITI)

17 PUBLICATIONS 143 CITATIONS

[SEE PROFILE](#)



Symeon Papadopoulos

The Centre for Research and Technology, Hellas

256 PUBLICATIONS 4,720 CITATIONS

[SEE PROFILE](#)



Ioannis (Yiannis) Kompatsiaris

The Centre for Research and Technology, Hellas

1,023 PUBLICATIONS 14,035 CITATIONS

[SEE PROFILE](#)



Pericles A. Mitkas

Aristotle University of Thessaloniki

320 PUBLICATIONS 3,022 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



InVID - In Video Veritas [View project](#)



KRISTINA: A Knowledge-Based Information Agent with Social Competence and Human Interaction Capabilities [View project](#)

MGraph: Multimodal Event Summarization in Social Media using Topic Models and Graph-based Ranking

Manos Schinas · Symeon Papadopoulos · Yiannis Kompatsiaris · Pericles A. Mitkas

Received: date / Accepted: date

Abstract Due to the increasing popularity of social media platforms, the amount of messages (posts) related to public events, especially posts sharing multimedia content, is steadily increasing. Sharing images can contribute to a rich and live coverage of the event. Yet, despite the value and interestingness of some posts, there is a lot of spam and redundancy, which makes it challenging to select the most important and characteristic posts for the event. In this work, we describe MGraph, a summarization framework that, given a set of social media posts about an event, selects a subset of shared images, simultaneously maximizing their relevance and minimizing their visual redundancy. MGraph employs a topic modelling technique based on different modalities to capture the relevance of posts to event topics, and a graph-based ranking algorithm to produce a diverse ranking of the selected high-relevance images. A user-centred evaluation on a dataset comprising a variety of real-world events demonstrates that MGraph considerably outperforms a number of state-of-the-art summarization algorithms in terms of relevance and diversity (25% and 7% improvement respectively).

Keywords Event Summarization · Social Media · Multimedia Ranking · Diverse Image Retrieval

This work was supported by the REVEAL project, partially funded by the European Commission, contract FP7-610928.

M. Schinas, S. Papadopoulos, Y. Kompatsiaris
Centre for Research and Technology Hellas (CERTH)
Information Technologies Institute (ITI)
Thessaloniki, 57001, Greece
E-mail: {manosetro,papadop,ikom}@iti.gr

Pericles A. Mitkas
Aristotle University of Thessaloniki
Department of Electrical and Computer Engineering
Thessaloniki, 54124 Greece
E-mail: mitkas@eng.auth.gr

1 Introduction

Due to their increasing popularity, microblogging platforms, and especially Twitter, have evolved into a powerful means for monitoring large scale public events. In such events, ranging from sports, to political events and festivals, event attendants typically capture and share their experiences through images and engage in discussions in social media. Thus, not surprisingly, the amount of event-related posts has reached impressive levels [1]. Importantly, **a growing number of these posts carry multimedia content, contributing to a rich and live coverage of the event**, since images typically convey a much more comprehensive impression of a specific situation compared to the limited text content of a microblogging post.

However, a significant percentage of posts can be considered as non-informative. Given the huge number of posts generated in the context of large events, this makes it very challenging to monitor the evolution of the event and understand its important moments. In the case of image sharing posts, the challenge stems from the abundance of images that carry little information about the event, e.g., memes, promotional banners, etc. In addition to irrelevant or low-quality content, there are considerable amounts of duplicate content in terms of text or visual appearance. Overall, **event-related streams of posts are highly diverse and noisy**, with different associated topics and conversations among users, and a high degree of redundancy. Thus, there is a profound need for event-based summarization mechanisms that can produce concise visual summaries, covering the main aspects of the event.

To this end, we propose **MGraph**, a graph-based framework that creates visual summaries of real-world events by post-hoc analysis of the stream of event-related

posts. MGraph leverages multiple modalities and signals associated with the posts. First, it computes the *significance* of each message, based on the *social attention* (i.e. the number of reposts) it receives. Then, it applies topic modelling to discover the underlying topics (aspects) of the event, and assigns messages to these topics. Next, it computes the *relevance* of each post with respect to its associated topic. In case of images, MGraph computes a *specificity* factor that penalizes images that are common across different event topics. Finally, the framework employs *DivRank*, a graph-based ranking algorithm, to obtain a set of relevant and significant posts that at the same time maximize the *coverage* of the event (by selecting the maximum possible number of topics) and minimize the visual redundancy among the selected images.

MGraph addresses multiple aspects of the summarization problem in a single framework. Through the multi-graph representation, which encodes different notions of similarity (textual, visual, temporal, social), the framework captures different modalities in social media posts, while the use of sophisticated graph-based methods, such as Clique Percolation for near-duplicate removal [27], SCAN [33] for topic detection, and DivRank [24] for diversity-oriented ranking, enables the extraction of high-quality visual summaries from massive amounts of event-related posts. To demonstrate the effectiveness of the proposed approach, we present a comprehensive evaluation on a reference dataset [22] consisting of numerous real-world events, and on two additional large event-focused datasets, and demonstrate that MGraph exhibits superior summarization performance in terms of precision and diversity compared to a number of state-of-the-art methods.

2 Related Work

2.1 Text-based Event Summarization

A substantial body of literature deals with the problem of textual summarization of microblogs, which is a special case of the *Multi-Document Summarization* (MDS) problem. One of the first MDS approaches relies on the computation of *centroids*, based on textual content. Then, the summary of a set of documents, represented by $tf \cdot idf$ vectors, consists of those documents that are closest to the centroid of the set [28]. Graph-based approaches have also been proposed to detect salient sentences from multiple documents, with *LexRank* [13] being the most notable among them: LexRank first constructs a graph of sentences (nodes), with the textual similarity between two sentences serving

as the connection (edge) between them. Then, it computes the *saliency* of each sentence using some centrality measure, such as the *Eigenvector Centrality* or its well known variant, the *PageRank* algorithm [26].

However, the text brevity, the informal writing and non-grammatical character of many microblogging posts, and the diversity of the underlying topics make the summarization problem much more challenging in the context of social media when compared to the standard MDS problem setting, where the input collection consists of long well-formed documents. In addition, the temporal dimension, which is a crucial element of microblogging posts, and the social interaction between users in social media platforms, are totally disregarded by previous MDS methods. To this end, numerous methods were proposed that incorporate not only the textual information of documents, but also their temporal and social dimension. The core idea of the majority of previous works is the clustering of documents set into coherent topics or sub-events and the selection of the most “representative” documents in each segment. Although there are works that investigate the use of social dimension to the problem of event detection [15], to our knowledge, this dimension is usually disregarded, compared to the number of methods that are based on content and temporal information.

Nichols et al. [25] describe a sports events summarization algorithm. This employs a *peak detection* algorithm to detect important moments in the timeline of tweets, and it then applies a graph-based technique to extract important sentences from the tweets around these moments. In [9], the authors propose a probabilistic model for topic detection in Twitter that handles the short length of tweets and considers time as well. Instead of relying only on the co-occurrences of words (as the majority of traditional probabilistic text models do), the proposed model uses the temporal proximity of posts to reduce the sparsity of the *term co-occurrence matrix*. Then, for each detected topic, the method considers the set of tweets with the highest similarity to the topic word distribution as the most representative. Shen et al. [30] present a participant-based approach for event summarization, which first detects the participants of the event, then applies a mixture model to detect sub-events at participant level, and finally selects a tweet for each detected sub-event based on the $tf \cdot idf$ centroid approach. In a similar work, Chakrabarti and Punera [7] propose the use of a *Hidden Markov Model* to obtain a time-based segmentation of the stream that captures the underlying sub-events.

Recent works focused on the creation of visual event summaries based on messages and content shared on social media. TwitInfo [21] is a system for summariz-

ing events on Twitter through a timeline display that highlights peaks of high activity. Alonso and Shiells [3] create football match timelines, annotated with the key match aspects, in the form of popular tags and keywords. Dork et al. [12] propose an interface for large events employing several visualizations, e.g., image and tag clouds. However, the aforementioned methods only make use of textual and social features for creating visualizations, and ignore the visual content of the embedded multimedia items.

2.2 Multimedia Event Summarization

The increasing use of multimedia content in microblog platforms has motivated numerous studies that consider visual information along with the textual content of microblog posts. Bian et al. [4] proposed a multimodal extension of *LDA* that detects topics simultaneously taking into account the textual and visual content of microblog posts with embedded images. The output of this method is a set of representative images for the underlying event. A slightly different problem is tackled by Lin et al. [18]. Unlike other methods that generate summaries as sets of posts or images, this method aims to create a storyline from a set of event-related multimedia objects. To this end, it constructs a multi-view graph of objects, with two types of edge, visual and textual, capturing the content similarity along with the temporal proximity among objects. Then, it extracts a time-ordered sequence of important objects by using Steiner trees [32].

The authors of [23] propose a method to select and rank a diverse set of images with a high degree of relevance to the event. A unique part of their work, is the use of external websites as sources of multimedia content, in cases where the amount of embedded images is insufficient for the creation of meaningful visual summaries. They use visual features first to discard irrelevant images and images of low quality, and then to detect and remove near duplicates among them to increase diversity. Then, they evaluate numerous ranking methods for selecting a small number of representative images for the event.

The majority of previous multimedia summarization approaches are mainly based on the textual and temporal information and ignore the richness of visual and social signals that are available in social media. To this end, the proposed framework, MGraph, incorporates textual, visual, temporal and social features to support the generation of visual summaries from event-focused collections of social media posts. MGraph extends our previous work [29] by performing a more comprehensive and extensive analysis, highlighting the role

and impact of individual components on the summarization performance of the whole framework. In particular, we present a number of additional studies and results exploring: a) the impact of the used topic modelling technique by comparing graph-based with probabilistic topic models, b) how the different elements of the proposed weighting scheme affect summarization performance, c) the effect of DivRank on the diversity of produced summaries, d) how MGraph benefits from the use of different modalities and how these modalities affect the ranking of images. In addition, we present experiments on two additional datasets around two large-scale events, namely the Baltimore Riots and the 2012 Presidential US Elections.

3 Framework Description

3.1 Overview

MGraph processes an *event-related set*¹ of social media posts to create a *visual summary* of the event. As visual summary, we define a set of images (contained in the set of posts) that are highly relevant to the event and capture the key moments of the event. As a first step, MGraph keeps only messages that are potentially *informative*. As *informativeness* is a rather subjective term and in many cases depends on the perspective of the end user, MGraph uses the *quality* of posts as a proxy of informativeness. To this end, the framework employs a set of filters to discard low-quality posts (subsection 3.3). Then, the framework builds a multi-graph to encode the similarity of posts across different modalities (subsection 3.4). Using this graph, it first detects and removes duplicates in terms of content (subsection 3.5) and it then detects the main event topics (subsection 3.6). Based on the extracted topic model, the framework computes a *selection score* for each message that captures the social attention that a message receives over time and the coverage of the corresponding topic (subsection 3.7). Finally, MGraph uses a graph-based ranking algorithm to diversify the images of the top ranked social media posts (subsection 3.8). An overview of the proposed method is illustrated in Figure 1. Note that, although the goal is to select a subset of images for the visual summary, the proposed framework makes use of all available social media posts, even those that do not carry any multimedia content.

¹ In microblogging platforms, such a set is typically formed by considering all posts that are tagged with an event-specific hashtag. In practice, despite being tagged with the event hashtag, many of these posts are irrelevant with the event, as in the case of promotional or trolling posts.

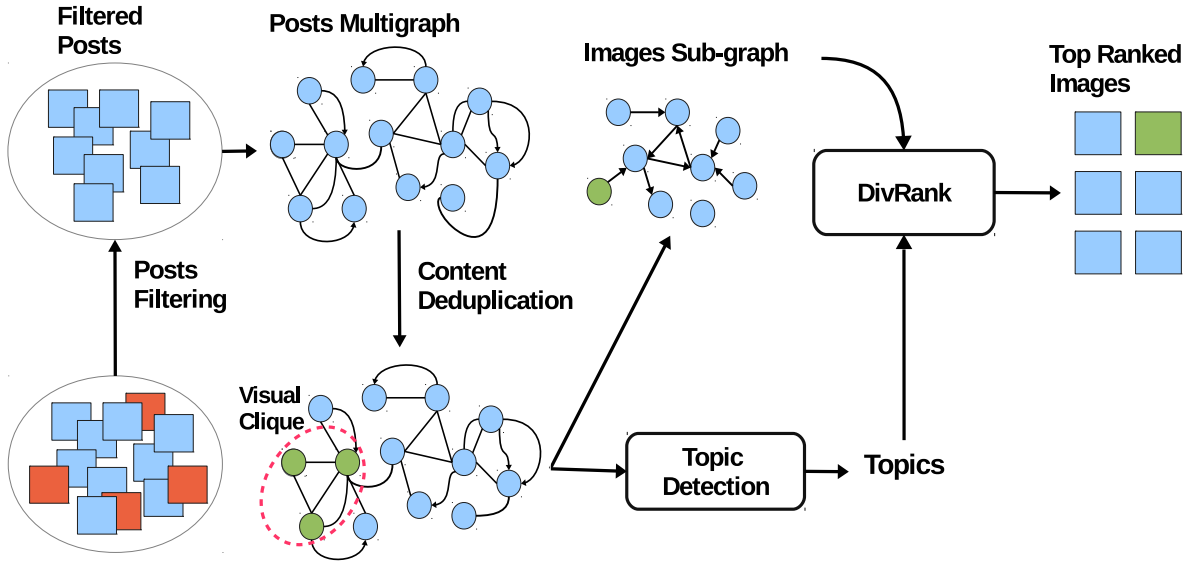


Fig. 1: Overview of MGraph framework.

3.2 Representation of social media data

The posts shared through social media platforms can be viewed as multimodal items. The three main dimensions include the *content*, *time* and *social interactions*. To capture these modalities, each post m is represented as a tuple $\{id, C, u, t_s, SI\}$, where id is a unique identifier of the post, C is its content, u an identifier of the posting user, t_s its publication time, and SI a tuple containing the associated social interactions.

The content C of a post consists of two elements: textual (C_{txt}) and visual (C_{vis}). C_{txt} is represented by a $tf \cdot idf$ vector v_m , where the tf part is the frequency of a term in the post text normalized by the maximum frequency in the post. Due to the short length of the documents in micro-blogging platforms, this component often equals to one and its contribution is limited. The inverse document frequency (idf) of each term is calculated over the whole set of posts. From the textual part of the content (C_{txt}), we also extract a set of detected named entities NE and a set of proper nouns PN . Note that we adapt $tf \cdot idf$ by using a constant boosting factor b to give more weight to named entities NE , user mentions m_u and proper nouns PN , since those are expected to be particularly relevant for the event. In other words, if a term w is a recognized named entity, its weight is given by $b \cdot tf_w \cdot idf_w$. The intuition is that two posts that share the same set of named entities and proper nouns or mention the same user, have a higher probability of relating to the same topic. The visual

part (C_{vis}) is optional, as not all items are associated with multimedia content. In case they are, we represent them using the combination of Speeded Up Robust Features (SURF) with the VLAD aggregation scheme [17], which was found to be a highly accurate and efficient visual feature representation [31].

Regarding the elements of the social interactions SI of a post, we consider the following three types: a) reposting of another post, b) replying to a post, and c) mentioning another user in the post text. Accordingly, SI consists of the following three elements: the id of the original post $refId$ in case the containing post is a repost of $refId$, an $inReplyId$ if the post is a reply to another one, and the set of mentioned users U_m .

3.3 Aggressive filtering

Content quality plays a key role in the generation of informative, but concise summaries. To this end, we first apply a set of heuristic rules to discard a significant amount of the initial set of event-related posts. In particular, we apply two types of filter on the posts. The first is based on the textual content and is applied on items that do not contain any embedded image. The second is based on the visual content and is applied only to posts with embedded images.

Text-based filtering employs a set of heuristic criteria to discard a post: a) very short text (e.g., less than six terms), b) inclusion of mentions to more than

three users, and c) inclusion of more than three URLs or hashtags. The core intuition behind these criteria is that posts of that type do not carry enough textual content to be usable in a summary, and that the combined use of URLs and hashtags or mentions is a strong indication of spam (an effort to direct users to the Web page pointed by the URL). Finally, in order to discard posts with incorrect or incomplete syntactic structure, we apply Part-Of-Speech tagging and keep only posts that match the regular expression of Equation 1, i.e. only items that contain at least one sentence that consists of at least a noun followed by a verb. Determiners and adjectives are optional.

$$(\text{determiner? adjective* noun+ verb})+ \quad (1)$$

Visual filtering is implemented by first discarding small images, i.e. images having width or height less than 200px, and then by discarding images of specific types that are typically inappropriate to be included in an event summary (even though the text of the containing post could be relevant to the event). Such images include memes, screenshots or images mainly comprising text. Figure 2 depicts such image examples coming from the Baltimore riots dataset. Although these images could be considered relevant in some contexts, they give no actual information or impression of the event. To discard this type of image, we first created classifiers for the following four classes: **meme**, **screenshot**, **heavy text**, and **real photo**. Each of those classifiers produces a *prediction score* $p_i \in [0, 1], i \in \{1, 2, 3, 4\}$, and we determine the class of the image as the one, of which the classifier produced the maximum prediction score, i.e. $i : \arg \max_i \{p_i\}$. In case the class is one of the first three, we discard the image, while we retain all images assigned to the class **real photo** for further processing. To build the four classifiers, we used the semi-supervised method of [20] and a hand-crafted training set consisting of approximately 900 manually selected Twitter images as positive examples for the first three classes and a random sample of 10,000 Flickr images from MIR-Flickr as positive examples for the **real photos** class [11, p.68]. Following the semi-supervised learning method of [20], we make use of the normalized VLAD vectors to extract a low-dimensional representation of the images, called Approximate Laplacian Eigenmaps (ALE) [14], which capture in a compact way the position of the image in the manifold of an underlying visual similarity graph (without actually constructing the graph). Then, using the set of labelled images and their ALEs as feature vectors, we trained an SVM classifier for each of the four aforementioned classes using a *one-vs-all* training scheme.

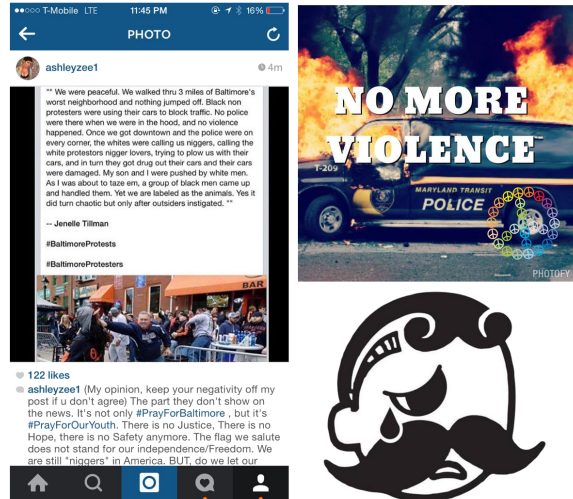


Fig. 2: Examples of relevant but non-informative images for the Baltimore riots (April, 2015).

3.4 Multi-graph generation

The remaining posts $M = \{m_1, m_2, \dots, m_n\}$ are used to construct a multi-graph:

$$\mathcal{G}_M = \{\mathcal{V}, \mathcal{E}_{txt}, \mathcal{E}_{vis}, \mathcal{E}_{soc}, \mathcal{E}_{time}\}, \quad (2)$$

where vertex $v_i \in \mathcal{V}$ corresponds to post m_i , and each type of edge corresponds to a different modality. \mathcal{E}_{txt} is a set of undirected edges expressing the textual similarity between nodes computed using the cosine similarity between the corresponding $tf \cdot idf$ vectors. \mathcal{E}_{vis} is a set of undirected edges that represent the visual similarity between posts containing images. This is the L_2 distance between the corresponding SURF+VLAD vectors. Note that we add an edge in \mathcal{E}_{txt} or \mathcal{E}_{vis} , only if the textual or visual similarity between the corresponding nodes is higher than θ_{txt} or θ_{vis} respectively. This thresholding operation aims to prune the graph, i.e. make it more sparse, and in that way to avoid adding spurious associations between nodes. The directed unweighted edges of \mathcal{E}_{soc} are based on the social interactions between users: we connect two posts m_i and m_j , with a directed edge from m_j to m_i , if post m_j is a direct reply to m_i , or m_j is a repost of m_i . Finally, the directed edges \mathcal{E}_{time} are derived based on the temporal proximity (TS) between posts m_i, m_j , published on t_i, t_j respectively, which is computed on the basis of the Gaussian kernel function of Equation 3.

$$TS(t_i, t_j) = \exp\left(-\frac{|t_i - t_j|^2}{2\sigma^2}\right), \quad (3)$$

where σ controls the spread of the sub-events within the main event. In general, the optimal value of σ depends on the type of event; in cases where sub-events

last longer, they require a higher value for σ and vice versa. For example a football match and the reactions of the viewers in social media may last some hours. On the other hand, events such as protests may last several days. The direction of an edge is from m_j to m_i , meaning that post m_j is published after m_i .

The multi-graph generation step requires the calculation of visual, textual and temporal similarity between all possible pairs of messages. The complexity of this step is $O(n^2)$, which makes it inapplicable to events that are associated with a very large number of social media posts. To reduce the complexity, for each post we efficiently retrieve its k nearest posts in terms of time, textual and visual content and then compute the pairwise similarities only between the post and the union of these three sets. Fast retrieval of top- k temporal neighbours takes place using *range queries* on a *B-Tree index*, retrieval of top- k textual neighbours using an implementation of *Locality Sensitive Hashing* for the cosine similarity metric [8], and retrieval of top- k visual neighbours using *Product Quantization* [16].

3.5 Content de-duplication

An important step for summarization that improves both the relevance and diversity of the produced summaries is the textual and visual de-duplication of content. In case of textual content an obvious source of redundancy is the reposting of messages. To this end, we keep only the original posts and discard all the explicit reposts. However, for each original post we also keep the count p of times it has been reposted by other users, and use it as a signal of the social attention it receives over time.

However, there are duplicates for which there is no explicit connection. This is more obvious in case of visual content, as users can post the same image or near-duplicates found on the Web or generated by them. To handle visual redundancy, we use the Clique Percolation Method (CPM) of [27]. In particular, we consider the sub-graph $\mathcal{G}_{vis} = \{\mathcal{V}, \mathcal{E}_{vis}\}$, keep only edges with visual similarity above a threshold θ_d , and use CPM to discover cliques of visual duplicates. Although we keep only visual edges corresponding to similar images in \mathcal{G}_M , introducing a higher threshold θ_d increases the likelihood that the remaining edges correspond to actual near-duplicate images. We represent the resulting cliques in a similar manner as single posts. More specifically, clique mc is a tuple $\{M_{mc}, C, t_s, p\}$, where M_{mc} is the set of posts in the clique, C denotes its aggregate content representation consisting of a textual and visual component, t_s is the mean value of publication times of the posts in M_{mc} , and p is the sum of re-posts

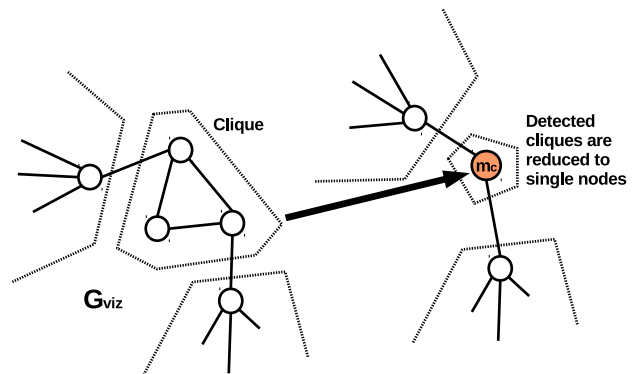


Fig. 3: Visual de-duplication using CPM.

of these posts. Regarding the textual part of C , we use a merged $tf \cdot idf$ vector v_{mc} (Equation 4). Contrary to $tf \cdot idf$ vectors of single posts, term frequencies in v_{mc} are important factors that express the importance of each term in the clique.

$$v_{mc} = \sum_{m \in M_{mc}} v_m \quad (4)$$

The aggregate visual representation of C is constructed using the SURF descriptors of all images in the clique and aggregating them in a single VLAD vector. To be more precise, we get each set of SURF descriptors extracted from each image in the clique and apply VLAD aggregation on the union of these sets. In this way, we take into account small variations between images (e.g., cropping, rotation, etc.).

After the clique detection step, we replace the clique posts in \mathcal{G}_M with the corresponding clique representations and re-calculate the corresponding edges of \mathcal{E}_{txt} , \mathcal{E}_{vis} , \mathcal{E}_{soc} and \mathcal{E}_{time} (Figure 3).

3.6 Topic detection

To detect the topics of an event, we opted for the Structural Clustering Algorithm for Networks (SCAN) [33]. SCAN is a graph clustering algorithm that is applied on a unified undirected unweighted graph $G = \{\mathcal{V}, \mathcal{E}\}$, where nodes correspond to the filtered set of event-related posts and cliques, and edges \mathcal{E} represent whether the two adjacent posts are of the same topic. To insert an edge in \mathcal{E} , we first check the existence of temporal (\mathcal{E}_{time}) and content-based edges (\mathcal{E}_{txt} , \mathcal{E}_{vis}). In other words, we connect two posts if they are close enough in time and have a high degree of content similarity at the same time. Apart from content similarity, we also use social interactions to add edges that increase the

density of inter-topic links. In particular, we connect two posts, regardless of their temporal proximity, if the one is a reply to the other, as the probability that these posts belong to the same topic is very high.

We apply SCAN on the resulting post graph, to identify dense sub-graphs of posts. These sub-graphs represent the underlying topics in the collection of posts: each topic is represented as a set of highly connected posts in the graph. Once the set of topics T is detected, we use the posts M_i associated with each $tp_i \in T$ to calculate a merged *tf · idf* vector v_i^{tp} that represents its textual content, in a similar manner to how we calculate merged vectors for cliques of posts. As images in a topic can be visually diverse, we do not compute a centroid representation for the visual content.

Using the SCAN algorithm for clustering results in a substantial amount of posts being kept outside of the detected clusters. These are divided into two categories, *hubs* and *outliers*. Hubs are nodes that are connected to more than one clusters, while outliers are nodes that are not connected to any of the clusters. Some of these posts can be considered as non-informative. However this is not the case for all of them, as some posts, despite not belonging to any cluster, may include valuable information that could attract a lot of social attention. This is more obvious in case of posts with images, which carry little textual content, and therefore typically have low textual similarity to other posts. Moreover, the visual appearance of images could be different, even for posts associated with the same topic. Therefore, it is likely that important images could be left out of the SCAN clusters. To this end, we do not discard the unassigned posts, but instead we form single-item clusters, both for hubs and outliers, and use them in the subsequent ranking process (subsection 3.7). In case of hubs, we also keep the number of communities to which a specific hub is connected and use it as a signal of the node *specificity* (to be discussed below).

3.7 Message selection and ranking

Our goal is to calculate an overall importance score for each of the posts and cliques to rank them and select the most representative ones. The importance score of a post m or clique mc is a combination of two factors: a) the *social attention* it receives over time, and b) the *significance of the topic* it is associated with.

Social Attention. The popularity of a post or clique, i.e. the number of the re-posts they receive over time, can be considered as a proxy of the social attention they receive. A high value of social attention often indicates an important and hence representative post regarding

the event. On the other hand, personal and other insignificant images, e.g. selfies, are expected to receive limited social attention. We measure social attention using Equation 5, where p is the number of re-posts and λ a smoothing parameter to prevent zero values of reposts. We opted for the use of a logarithmic function due to the fact that the number of re-posts in social media follows a highly skewed distribution as stated in recent works ([19], [35]). In this way, extremely large numbers of reposts are normalized.

$$S_{att}(m) = \log_2(p + \lambda) \quad (5)$$

Topic Coverage. The association of a post with a detected topic could be a strong indication of its importance, since that post would be expected to contribute valuable information about a specific aspect of the event. However, some posts are more representative of a topic compared to others. Moreover, some topics are more significant than others, and accordingly posts associated with these topics should receive higher importance scores. To this end, we assess the topic coverage of a message using Equation 6. Its first part captures the relevance to the topic and is calculated as the textual similarity of post m to the topic centroid v_i^{tp} . Its second part captures the significance S of the underlying topic, so that posts from larger and denser clusters receive higher scores. To measure the density D_i of topic tp_i , we use the corresponding sub-graph detected by SCAN, and the number of edges $|E_i|$ and nodes $|V_i|$ in it. Density is a measure of the generality of a given topic. A topic that corresponds to a sub-graph of high density is typically focused (specific), since posts in this topic have a high degree of content similarity with each other. On the other hand, a topic with low density is considered to be generic (i.e. to lack focus), therefore posts associated with it are considered less informative and hence should be penalized.

$$S_{cov}(m) = \cos(u_m, v_i^{tp}) \cdot S(tp_i) \quad (6)$$

$$S(tp_i) = D_i \cdot \exp\left(\frac{|M_i|}{\max_{k \in T} |M_k|}\right) \quad (7)$$

$$D_i = \frac{2|E_i|}{|V_i||V_i| - 1} \quad (8)$$

The overall significance score of a message or clique is the product between its social attention and the respective topic relevance (Equation 9).

$$S_{sig}(m) = S_{att}(m) \cdot S_{cov}(m) \quad (9)$$

3.8 Image Ranking and Diversification

The motivation behind computing the importance score of Equation 9 is to generate a set of relevant high-quality images in the top ranked positions of the summary. However, there are images that are considered very popular, but they are not highly relevant to the event of interest. For example, an image depicting the flag of Ukraine could be considered to be related to an event about the Ukraine crisis, but it does not provide important information about the event. To address this shortcoming of the overall score, we introduce a *specificity* factor that penalizes such images.

Image specificity is a measure of how much information an image provides for a specific event, i.e. whether the image is common across all topics of the event. We calculate image specificity S_{spec} for each image I using the *idf*-like score of Equation 10.

$$S_{spec}(I) = \log\left(\frac{|T|}{|T_I|}\right), \quad (10)$$

where $|T|$ is the number of topics in the event and $|T_I|$ is the number of topics containing image I . We calculate $|T_I|$ in the following way. We first reuse the de-duplication technique presented in subsection 3.6 to measure the number of topics $|T_I|$ that contain an image I or its (near) duplicates.

For the rest of the images that do not form visual cliques we check whether they are contained in the set of hubs detected by the SCAN algorithm. More specifically, for an image of which the containing post is identified as hub, we measure the number of communities to which it is connected. However, to consider such a connection as valid and take it into account we demand that the maximum visual similarity to the members of the community exceeds the same threshold θ_d used by the CPM method. For example, Figure 4 depicts a hub H that is connected to three communities. While there are three adjacent communities only in two of them there is at least a visual edge above threshold θ_d (blue lines). In case of community 3, both edges have a weight below θ_d (red lines). To this end, we consider as valid only the connections to communities 1 and 2.

Finally, the image selection score $S(I)$ of image I is the product of the importance score (Equation 9) and the image specificity score. As a result, images that depict the same aspect of the event and have high visual similarity to each other, may have similar selection scores. This is more obvious for posts that have limited social attention but are part of large (important) topics. In that way, plain use of the selection score to rank images would result in visually redundant images becoming part of the summary. To incorporate diversity into

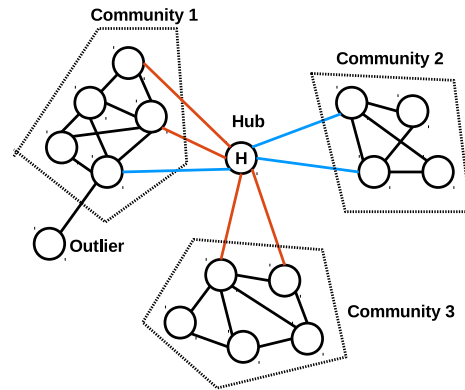


Fig. 4: Hub specificity calculation. Blue lines indicate visual edges with weight exceeding θ_d , while red lines correspond to visual edges below this threshold.

the score calculation, we employ DivRank [24], a variant of PageRank that aims to maximize diversity. The intuition behind DivRank is that nodes related to other significant nodes should be ranked higher, but inside dense sub-graphs of the graph only a single node should be promoted, while the rest of the highly-connected nodes should be penalized.

To apply DivRank, we use the initial multi-graph \mathcal{G}_M to generate a directed sub-graph $\mathcal{G}_V = \{\mathcal{V}_V, \mathcal{E}_V\}$. Vertices $\mathcal{V}_V \subset \mathcal{V}$ are the subset of posts that contain an embedded image and will be candidates for the visual summary. For the creation of set \mathcal{E}_V , we combine the two sets \mathcal{E}_{vis} and \mathcal{E}_{time} . In particular, for each pair of vertices $v_i, v_j \in \mathcal{V}_V$, we create a weighted directed edge $e \in \mathcal{E}_V$ with the same direction as the corresponding edge in \mathcal{E}_{time} . The weight of this edge is the visual similarity between the adjacent vertices. Note that for pairs of images that do not share a temporal edge $e_t \in \mathcal{E}_{time}$ there is also no connection in \mathcal{G}_V . Based on this, the resulting summary is not only expected to be diverse in terms of visual content but also in terms of time. This feature is quite important especially for large-scale events that last many hours or days, enabling the selection of images during the whole duration of the event in a principled manner. To ensure convergence for DivRank, we normalize the edge weights, such that the sum of the adjacent out-edges of each post equals to one. To calculate the new selection score, we apply DivRank using the iterative scheme of Equations 11 and 12.

$$\mathbf{r} = dW^{-1}\mathbf{r} + (1-d)\mathbf{h} \quad (11)$$

$$W = dW\mathbf{r} + (1-d)\mathbf{h} \quad (12)$$

Vector \mathbf{r} holds the DivRank scores and d is a dumping factor that controls the impact of the initial score to the re-ranking process. The initial value of matrix W is the adjacency matrix derived from the directed graph \mathcal{G}_V . Also, instead of using a uniform value for the priors \mathbf{h} , we use the value of the calculated score of each image in the graph. Specifically, the prior \mathbf{h}_i of node i in the graph that corresponds to image I_i is $\mathbf{h}_i = S(I_i)$.

4 Evaluation

4.1 Dataset and experimental setting

To evaluate the proposed framework, we conducted a set of experiments in three different data sets. The first one is the dataset of McMinn et al. [22] that contains tweets for more than 500 events of different types. We used the 50 largest events in terms of number of tweets, as in the work of McParlane et al. [23]. These events range from sports events, e.g., the Sochi winter Olympics, to political events such as the Ukraine crisis and the Venezuelan protests. The dataset contains 364,005 tweets in total, while each event is associated with 4,730 tweets on average. However, due to suspended accounts and deleted messages we managed to fetch only 296,160 of these tweets. About 3,51% of these, i.e. 12,772 tweets, contain an embedded image. The second data set is related to the riots in Baltimore that followed Freddie Gray’s hospitalization and subsequent death in police custody. The dataset consists of 1,281,883 tweets containing the hashtags `#BaltimoreRiots`, `#BaltimoreProtests`, `#FreddieGray`, and `#BaltimoreUprising`, which were intensively used during the event. The third dataset is the one created and used for evaluating a number of Twitter topic detection methods [2] and consists of 1,106,712 tweets related to the 2012 US presidential elections. Table 1 shows some basic statistics for the second and third datasets.

In [23], the authors used CrowdFlower² to create relevance judgements for the top five images selected for summarization for each of the 50 events. This resulted in the generation of relevance annotations for a very small percentage of the images in the dataset. To this end, we follow the same approach as [23] to create relevance judgements, in a scale from 0 (not relevant) to 3 (relevant), for the union of images selected as summaries by all the methods used in the evaluation (to be described below). In order to study more comprehensively the performance of methods, we selected 20 images for each of the 50 events of [23]. For the datasets of Baltimore Riots and US elections we performed a

Table 1: Datasets for Baltimore riots and US elections. `#Accepted` denotes the number of tweets that were not discarded from the filtering steps of subsection 3.3. The numbers for `#Users`, `#Replies`, `#Images` and `#Uniq. Images` refer to the `#Accepted` tweets.

	Baltimore Riots	US Elections
<code>#Tweets</code>	1,281,883	1,106,712
<code>#Original</code>	266,712	791,933
<code>#Accepted</code>	214,142	440,621
<code>#Users</code>	103,677	303,415
<code>#Replies</code>	20,959	23,690
<code>#Images</code>	26,834	13,645
<code>#Uniq. Images</code>	18,589	12,784

more extensive evaluation by creating relevance judgements for all contained images. We asked from a group of human annotators to evaluate how relevant and representative are the selected images to the corresponding event. We ensured that each pair received three judgements at least, from different users. The group of annotators comprised 25 persons 24-32 years old, educated in the field of computer science, having some experience in the use of Twitter and social media. The task given to annotators was the following:

Task Description

You are presented with an image and an event title (describing a “trending” topic in Twitter). For each image and event title, you are asked to answer the following question:

Question: Is this image relevant to the event?

Possible Answers:

0. *The image is clearly not relevant to the event.*
1. *The image is probably not relevant to the event, but I am not entirely sure.*
2. *The image is somewhat relevant to the event, but I have my doubts on whether I would like to see it in a photo coverage of the event.*
3. *The image is clearly relevant to the event, and I would like to see it in a photo coverage of the event.*

We used several open-source libraries to analyse the text of the tweets. For tokenization we opted for the `StandardAnalyser` provided by Lucene, which performs well in English text. For named entity detection we used the Stanford NER library with the default 3-class model. For Part-of-Speech (POS) tagging we used the Stanford POS Tagger, but opted for the Twitter-specific POS model from the ARK research group³. For visual features, we extracted Speeded Up Robust Features (SURF) from each image of the dataset using

² <http://www.crowdfunder.com/>

³ <http://www.ark.cs.cmu.edu/TweetNLP>

the implementation of [31]⁴. We then used four code-books of 128 visual words (in total 512) to quantize each descriptor and used the VLAD scheme to aggregate the descriptors of each image into a single vector of $64 \cdot 512 = 32,768$ dimensions. Finally, we used PCA to create a 1024-dimensional $L2$ -normalized reduced vector that represents the visual content of the image.

For the generation of multi-graph \mathcal{G}_M , we retrieve the $k = 500$ nearest neighbors of each message in terms of textual, visual and temporal similarity (1,500 maximum in total, since there were overlaps among the three sets). The visual and textual similarity thresholds were empirically set to $\theta_{vis} = 0.5$ and $\theta_{txt} = 0.6$ respectively. Parameter σ^2 of the temporal kernel was empirically set to 24 hours as most of the important sub-events in the first dataset last less than a day. In other words, the temporal proximity between tweets in the same day is more than 0.6. For visual de-duplication, threshold θ_d was set to 0.65 which corresponds to images that are near-duplicates in terms of visual content. In the topic detection step, we set the parameters of SCAN to $\mu = 2$ and $\epsilon = 0.65$. Finally, in the ranking step with DivRank we set $d = 0.75$ to most of the experiments. However, we also conducted an experiment to investigate the effect of this factor in the results. We make all datasets, relevance judgements, and source code of the implementations used in the experiments publicly available⁵.

4.2 Evaluation metrics and baselines

We applied the proposed method (denoted as **MGraph**) to the dataset tweets to generate a representative summary for each of the contained events. In particular, we ranked the images according to their DivRank score and kept the top N as the summary. N can vary and depends on the initial size of the event-related posts. In general, it can be set by using a compression rate, e.g. N corresponds to the top 1% of the posts related to the event. In our case we used values of N equal to 1, 5, 10 for small events, while we expanded this to $N = 100$ and $N = 500$ for large scale events. As relevant we considered posts with images that were annotated on average with a score equal to or larger than 2 in the [0-3]-scale presented above. We evaluated the average performance of our method in a similar manner as [23] by calculating the following metrics:

- **Precision (P@N)**: The percentage of images among the top N that are relevant to the corresponding event, averaged among all events. We calculate precision for N equal to 1, 5, and 10.
- **Success (S@N)**: The percentage of events, where there exist at least one relevant image amongst the top N returned, for $N=10$.
- **Mean Reciprocal Rank (MRR)**: Computed as $1/r$, where r is the rank of the first relevant image returned, averaged over all events. In case of a single event (as in the cases of the Baltimore Riots and US Elections datasets), the metric reduces to Reciprocal Rank (RR).
- **α -normalized Discounted Cumulative Gain**: Clarke et. al. [10] proposed a modified version of nDCG, called α -nDCG, for evaluating novelty and diversity in search results. α -nDCG discounts gains not only based on the rank of a document as in traditional nDCG but also based on the *information nuggets* already seen. In our case, as information nuggets we consider the topics of the event. The gain of each image is based on the annotation options in the [0,3] scale. Parameter α is set to 0.5 to keep a balance between relevance and diversity.
- **Average Visual Similarity**: AVS@N measures the average visual similarity among all pairs of images in the top N selected images, averaged over all events. Lower AVS values are preferable since they imply higher diversity in terms of visual content.

We compare the proposed **MGraph** framework with several image ranking methods. Note that we applied the same filtering and de-duplication steps to all methods. More specifically, we evaluated the following summarization methods:

- **Random**: selects N random posts with images from the (filtered) set of images as the summary set.
- **MostPopular**: picks up the most popular posts with images in terms of re-posts. This corresponds with ranking based on the score of Equation 5.
- **LexRank**: uses the graph $G = \{\mathcal{V}, \mathcal{E}\}$ of subsection 3.6 to rank nodes based on the LexRank algorithm [13], and selects the top N nodes with images.
- **TopicBased**: selects the most relevant posts from the most significant topics according to the score of Equation 6.
- **P-TWR**: ranks images in descending order using the weighting scheme described in [23].
- **S-TWR**: groups the posts of each event into semantic clusters and selects the top ranking of each using the weighting scheme of [23]. For the dataset of [22] we used the clustering provided by the authors of [23]. For the other two datasets, we used the same SCAN-based clustering described in subsection 3.6.

⁴ <https://github.com/MKLab-ITI/multimedia-indexing>

⁵ <https://github.com/MKLab-ITI/mgraph-summarization>

Table 2: Comparison of summarization methods in terms of precision, averaged among all events in the dataset of McMinn et al. [22]. Bold values indicate the highest performing method for the given metric.

Method	P@1	P@5	P@10	S@10	MRR
Random	0.391	0.400	0.405	0.800	0.562
MostPop	0.522	0.469	0.446	0.848	0.669
LexRank	0.456	0.452	0.420	0.847	0.611
TopicBased	0.457	0.473	0.469	0.847	0.620
P-TWR	0.521	0.486	0.437	0.826	0.673
S-TWR	0.478	0.452	0.435	0.869	0.661
MGraph	0.587	0.518	0.544	0.913	0.728

4.3 Results

Table 2 contains several precision-oriented metrics for both MGraph and the competing methods for the dataset of [22]. Not surprisingly, the worst results for all the metrics are those of the **Random** selection. Regarding P@N the best results were achieved from MGraph. For P@1, popularity-based methods, such as **MostPopular** and **P-TWR**, achieved very good results as would be expected. This means that the image having the highest value of popularity, has a higher possibility of being relevant to the event. However, the performance of these two methods drops significantly for P@5 and P@10. This is explained by the fact that although some image might be considered to be irrelevant, it could still attract attention for a number of other reasons (e.g., it could be funny), and would therefore be highly ranked by popularity-based methods. Success for the top 10 retrieved images is high for all methods, even for the **Random** one. However, even in this case our method outperforms all others in terms of S@10.

The average Mean Reciprocal Rank (MRR) is also higher for our method, with popularity-based methods achieving the next best results. Note that the average performance for this metric for popularity-based methods benefits from cases where the most popular image is relevant. This mainly occurs when the number of reposts of an image gets extremely high values, e.g. hundreds or thousands of reposts. For example in the US Elections dataset the most shared post was that of Barack Obama depicted in Figure 5. This tweet got a huge number of retweets, while the same picture was tweeted by different users independently. Thus, the clique that represents this image was ranked in the first position by all methods that use some measure of popularity. However, in events that there are no such images, the performance drops significantly. In contrast, our method handles successfully such cases, as it does not solely rely on the popularity of images, but also considers their association with the underlying topics.

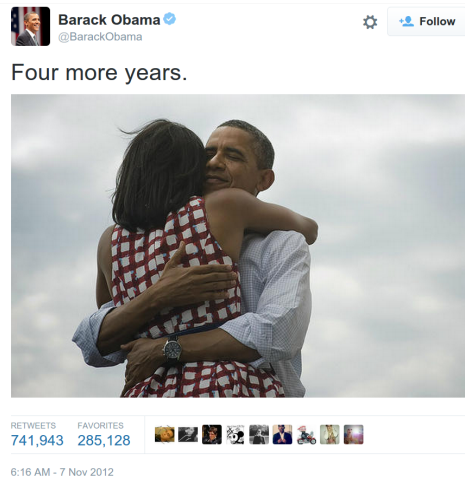


Fig. 5: “Four more years.”: the most shared image in the US Elections dataset promoted in the first positions from all popularity-based methods.

Table 3 presents a comparison among methods in terms of their diversity performance. According to it, MGraph achieves the best value of α -nDCG@10, with S-TWR having the second best performance. This indicates that the use of the DivRank algorithm resulted in a more diverse set of relevant images compared to other methods. Compared to S-TWR that aims to achieve diversity through image clustering, our method achieves a 7% improvement in terms of α -nDCG. It is noteworthy that this improvement is achieved without sacrificing precision, as P@10 compared to S-TWR is also improved by 25%. In case of average visual similarity between images the best result is obtained by S-TWR.

Our method has somewhat worse performance in terms of AVS@5, where it is ranked second, while for AVS@10, it is ranked third. The worst results in terms of AVS are obtained using LexRank. This is reasonable as LexRank is based on the PageRank algorithm, and hence it favours images that are highly connected, i.e. images that are highly similar in terms of visual content. One should be cautious regarding the interpretation of AVS: although it is a reasonable measure of diversity, it is solely based on the use of visual features, hence it might not be able to capture users’ perception and experience. In addition, it is expected that the inclusion of irrelevant images in the set of selected, would result in lower values for AVS, but this is obviously not desirable.

The events in the dataset of [22] belong to six categories, as shown in Table 4. Each of these categories has different characteristics resulting in variations in the performance of our method. For example, the *Arts & Entertainments* category is more prone to duplicate messages and images, e.g., tweets with images of celebri-

Table 3: Comparison of summarization methods in terms of diversity, averaged among all events in the dataset of McMinn et al. Bold values indicate the highest performing method for the given metric.

Method	α -nDCG@10	AVS@5	AVS@10
Random	0.657	0.024	0.019
MostPop	0.717	0.022	0.018
LexRank	0.685	0.081	0.056
TopicBased	0.689	0.035	0.027
P-TWR	0.717	0.020	0.016
S-TWR	0.722	0.011	0.010
MGraph	0.774	0.018	0.021

Table 4: MGraph performance across event categories.

Category	P@10	α -nDCG	AVS@10
Law & Politics	0.536	0.729	0.047
Arts & Entertainment	0.700	0.721	0.048
Science & Technology	0.800	0.896	0.059
Disasters & Accidents	0.450	0.492	0.013
Sports	0.500	0.624	0.025
Miscellaneous	0.368	0.606	0.053

ties. The best P@10 measure was obtained for events about *Science & Technology*, but this should be taken with caution, as this category contains very few events. The second best P@10 score was obtained for events about *Arts & Entertainment*. This can be explained by the fact that these events refer mostly to celebrities and the corresponding images usually depict them in a manner that is relevant to the event. Regarding average visual similarity, the best value is achieved for events about *Disasters & Accidents*. This is can be explained taking into account that images of such events, e.g., earthquakes, can be quite diverse in terms of their appearance even in cases they refer to the same event.

4.3.1 Summarization of large-scale events

As mentioned before, in McMinn’s dataset [22] each event is associated with 4,730 tweets on average. To gain further insights into the performance of MGraph on large-scale events, we conducted a set of experiments on two additional datasets. Such events, which typically last many days, consist of many sub-events and the number of associated posts is at least some hundreds of thousands; hence, summaries of 10 or 20 images are not sufficient. For this reason, we created much larger summaries of $N = 100$ and $N = 500$ images and calculated the same evaluation metrics as above.

Table 5 contains the precision-oriented metrics for both MGraph and the competing methods for the Baltimore Riots dataset. MGraph achieves superior performance as the precision remains equal to 1, even for $N = 100$ and to 0.988 for $N = 500$. This means that

Table 5: Comparison of summarization methods in terms of precision for the Baltimore Riots data set. Bold values indicate the highest performing method for the given metric.

Method	P@10	P@100	P@500	RR
Random	0.400	0.500	0.440	0.333
MostPop	0.500	0.530	0.542	1.000
LexRank	0.800	0.550	0.516	1.000
TopicBased	0.700	0.560	0.562	1.000
P-TWR	0.700	0.400	0.566	1.000
S-TWR	0.800	0.820	0.586	0.500
MGraph	1.000	1.000	0.988	1.000

Table 6: Comparison of summarization methods in terms of diversity for Baltimore Riots data set. Bold values indicate the highest performing method for the given metric.

Method	α -nDCG@100	AVS@10	AVS@100
Random	0.411	0.112	0.151
MostPop	0.737	0.081	0.070
LexRank	0.651	0.294	0.162
TopicBased	0.880	0.020	0.055
P-TWR	0.723	0.012	0.021
S-TWR	0.781	0.011	0.041
MGraph	0.882	0.018	0.061

the image summaries created by MGraph consist of images that are either relevant or somewhat relevant to the event. Note that for $N = 10$, all methods, except Random, achieve high precision. The Reciprocal Rank is also high, meaning that most methods succeed in ranking a relevant image at the first place. Regarding diversity (Table 6), MGraph achieves the best results in terms of the α -nDCG@100 metric. In terms of AVS, the best results are obtained by S-TWR and P-TWR. LexRank has again the worst performance in terms of AVS.

Similar conclusions can be drawn from the results on the US Elections dataset. MGraph achieves superior performance in this dataset in terms of precision-oriented metrics. In contrast, MGraph does not achieve the best performance for any of the diversity-oriented metrics. We assume that this is caused mainly by the fact that, in this dataset, the constructed content-based graphs are sparser compared to the rest of the datasets. As a result, SCAN and DivRank fail to create high-quality topics and ranking. Note that in terms of α -nDCG@N, MGraph has the second best value, which is comparable to the best value obtained by the TopicBased approach. We believe that as α -nDCG@N is a balanced measure between precision and diversity, in this case the achieved value of MGraph benefits from its excellent performance in terms of precision.

Table 7: Comparison of summarization methods in terms of precision for the US elections data set. Bold values indicate the highest performing method for the given metric.

Method	P@10	P@100	P@500	RR
Random	0.500	0.530	0.546	1.000
MostPop	0.700	0.590	0.558	0.500
LexRank	0.800	0.690	0.684	1.000
TopicBased	0.900	0.870	0.738	1.000
P-TWR	0.800	0.590	0.564	0.333
S-TWR	0.900	0.580	0.562	0.500
MGraph	1.000	1.000	1.000	1.000

Table 8: Comparison of summarization methods in terms of diversity for the US elections data set. Bold values indicate the highest performing method for the given metric.

Method	α -nDCG@100	AVS@10	AVS@100
Random	0.508	0.113	0.092
MostPop	0.710	0.089	0.097
LexRank	0.589	0.121	0.229
TopicBased	0.802	0.077	0.081
P-TWR	0.777	0.067	0.094
S-TWR	0.790	0.052	0.041
MGraph	0.801	0.101	0.114

4.3.2 Precision-Recall analysis

To examine the performance of the proposed method in terms of recall, in other words to show that **MGraph** does not miss relevant images, we create Precision-Recall (P-R) interpolated curves for the Baltimore Riots and US Elections datasets. The curves are illustrated in Figures 6 and 7 respectively. Note that although most methods give good results in terms of P@N for $N = 10$, $N = 100$ and $N = 500$, these values of N correspond to the low range of values for Recall. For example, for $N = 500$ the best possible recall values are $r = 0.048$ and $r = 0.1$ in the Baltimore Riots and US Elections datasets respectively. For the largest part of P-R curves, and especially for $r > 0.1$, most of the methods exhibit only slightly better performance compared to **Random**. In other words, although all the methods achieve to rank relevant images to the first positions of the summary, performance drops significantly for positions lower in the ranking. In contrast, **MGraph** achieves a remarkably better ranking as precision remains high even for recall values up to 0.2. Beyond that value, precision starts to drop as irrelevant images start to be erroneously included to the summary.

4.3.3 DivRank performance analysis

We also study how parameter d of DivRank affects the precision and diversity of **MGraph**, testing different val-

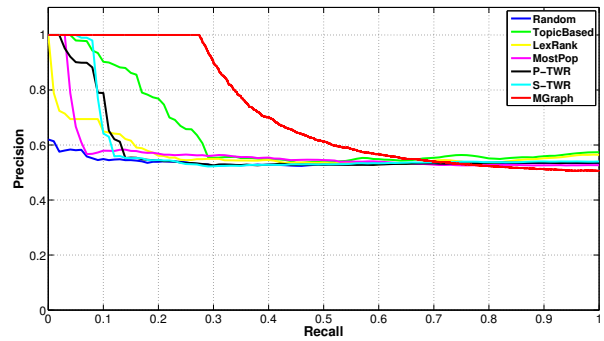


Fig. 6: P-R interpolated curve on Baltimore Riots.

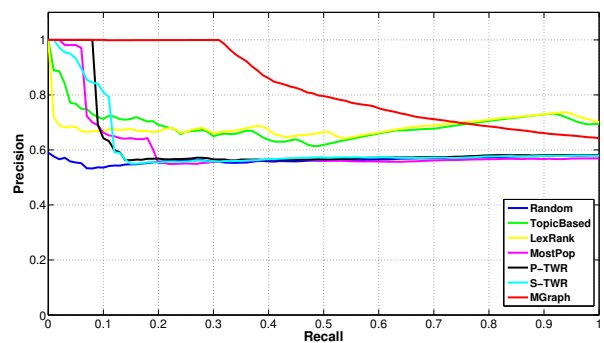


Fig. 7: P-R interpolated curve on US Elections.

ues from 0 to 1, and calculating P@10, S@5, MRR and α -nDCG@10 for each of them. The results for the dataset of [22] are depicted in Figure 8. The worst results for all metrics are obtained for $d = 0$. Essentially, in this marginal case, the re-ranking procedure of DivRank is not performed as the first part of Equations 11 and 12 is equal to zero. The best results are achieved for $0.7 \leq d \leq 0.8$, but even for $d > 0.8$ the performance remains almost steady for most of the metrics. The slight decrease for $d > 0.8$ can be explained by the fact that for such extreme values of d , DivRank attempts to create a more diverse set of images, which makes it more likely to introduce irrelevant images in the top ranks of the result set.

4.3.4 DivRank versus Maximal Marginal Relevance

To comparatively evaluate the performance of DivRank, we also used a greedy approach for re-ranking based on Maximal Marginal Relevance (MMR) [6]. In particular, we rank images from the higher to the lower selection score $S(I)$ and add them in the summary under the constraint that visual similarity to each image in the set of already selected images is below a threshold. Redundant images are discarded from the ranking.

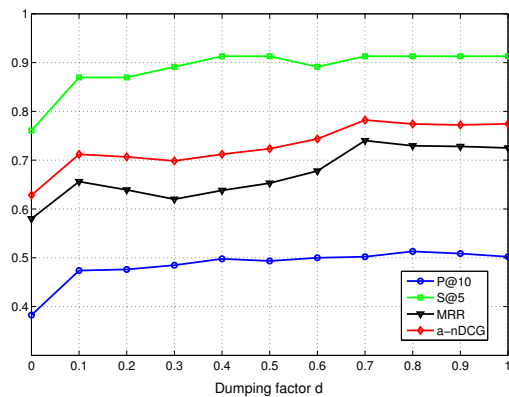


Fig. 8: Effect of dumping factor d on P@10, S@5, MRR and α -nDCG@10, in McMinn’s dataset.

Table 9: Comparison between DivRank and MMR, in McMinn’s dataset ($N = 10$).

Method	P@N	AVS@N	α -nDCG@N
DivRank	0.544	0.021	0.744
MMR	0.491	0.010	0.711

With this approach visual redundancy is minimized, while images in the summary have the highest possible score. Note that, with the current implementation of SURF+VLAD, images with visual similarity above 0.65 are considered as duplicates. Similarities between 0.4 and 0.65 correspond to similar but not identical images, and finally, visual similarities below 0.4 correspond to dissimilar images. For this reason, the threshold for MMR was empirically set to 0.4.

According to Table 9, DivRank achieves better results in terms of precision ($P@10$) and α -nDCG@N. This is explained by the fact, that DivRank is based on a graph that captures the visual similarity between images in a higher order than MMR that only compares pairs of images. In other words, although DivRank promotes diversity, at the same time it can identify cases of isolated and insignificant nodes and penalize them. On the other hand, MMR just compares an image with the set of already selected images and does not take into account the position of an image in the visual graph. Not surprisingly, MMR leads to better results in terms of AVS@10 as it explicitly discards similar pairs of images. Regarding α -nDCG@N, DivRank achieves also a slightly better value. The explanation for this is related again to the use of the underlying graph in the ranking process: As α -nDCG@N discounts the gain of images from the same topic, DivRank is expected to perform better due its tendency to promote images from different areas on the graph, which typically correspond to different topics.

4.3.5 Impact of modalities

As discussed in subsection 3.4, multi-graph G_M captures the similarity between two posts across different modalities. In that way, we should expect that topic detection and ultimately summarization performance will be affected by the modalities used. Figure 9 illustrates one of the detected sub-graphs in the Baltimore Riots dataset, using different colors to depict the different types of similarity (computed using different modalities). It is obvious that excluding any of the modalities would cause important changes in the graph structure. To quantify the impact of such changes, we conducted a set of experiments, in which we removed the corresponding type of similarity from G_M and evaluated the performance of MGraph. Table 10 presents the results in terms of precision and diversity for instances of MGraph without visual, social and temporal associations, which are denote as $MGraph_{NV}$, $MGraph_{NS}$ and $MGraph_{NT}$ respectively. On the dataset by McMinn, performance drops, but to a limited extent in most cases. Regarding precision ($P@10$), all instances perform worse than the original version of MGraph, but still better than most competing methods. The same holds for the MRR metric. Overall, the most pronounced negative impact on precision is caused by the removal of temporal edges. For diversity-oriented metrics, the most noticeable drop in performance occurred for $MGraph_{NV}$, which was expected since this instance does not make use of the visual similarities between images.

Although the average performance of MGraph instances does not change remarkably, more careful examination of the results reveals that performance drops mostly for the largest events in McMinn’s dataset. This is reasonable, as for small events with few posts and limited diversity, the structure of the multi-graph M_G does not show important variations as a result of adding new modalities. In contrast, in large-scale events with many posts (nodes), the multi-graph structure could be significantly different as a result of adding edges based on a new modality. This observation is attested by the results for the other two datasets, which correspond to large-scale events. In these two datasets, visual and social associations have important contributions to the structure of M_G . More specifically, the graph constructed from the Baltimore Riots dataset has 214,142 nodes (posts) and 6,750,635 textual edges, 2,248,810 visual edges, and 19,144 social interaction edges. Thus, the visual and social edges account for 24% and 0.2% of the total edges respectively. It is therefore expected that removing social edges does not have an important impact on the overall performance of the framework.

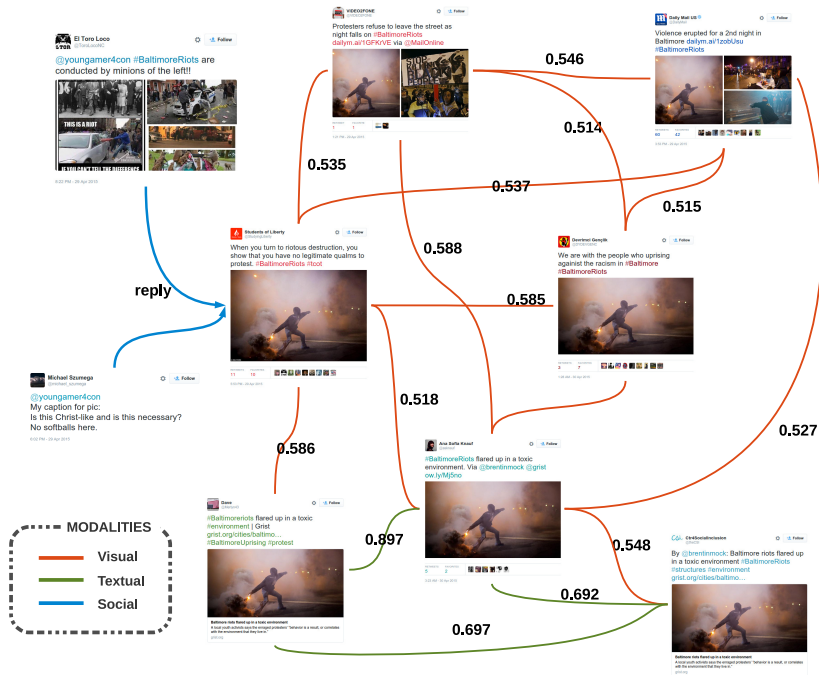


Fig. 9: Example of multi-graph in Baltimore Riots dataset (graph density is equal to 0.5).

Table 10: Impact of modalities on summarization. Bold values indicate the highest performing method for the given metric.

McMinn’s dataset, $N = 10$				
Method	P@N	MRR	α -nDCG@N	AVS@N
MGraph	0.544	0.728	0.774	0.021
MGraph _{NV}	0.522	0.720	0.688	0.055
MGraph _{NS}	0.540	0.718	0.770	0.021
MGraph _{NT}	0.491	0.690	0.751	0.022

Baltimore Riots dataset, $N = 100$				
Method	P@N	RR	α -nDCG@N	AVS@N
MGraph	1.000	1.000	0.882	0.061
MGraph _{NV}	0.820	1.000	0.797	0.245
MGraph _{NS}	0.950	1.000	0.891	0.059
MGraph _{NT}	0.710	0.500	0.832	0.090

US Elections dataset, $N = 100$				
Method	P@N	RR	α -nDCG@N	AVS@N
MGraph	1.000	1.000	0.801	0.114
MGraph _{NV}	0.600	1.000	0.591	0.454
MGraph _{NS}	0.970	1.000	0.802	0.112
MGraph _{NT}	0.830	1.000	0.704	0.097

Overall, the results in Table 10 support the original hypothesis that summarization benefits from the use of multiple modalities. The original version of MGraph that takes into account all modalities to build the multi-graph G_M achieves the best results compared to its in-

stances that miss one of the modalities. This observation is confirmed in all three datasets in terms of precision. However, not each modality has the same impact on summarization. The exclusion of social connections seems to have only minimal negative effect on the results. On the other hand, visual and temporal edges appear to be important for retaining high summarization precision. Diversity-oriented metrics are affected mostly from the exclusion of visual information.

4.3.6 Impact of weighting components

The proposed weighting scheme is quite complex as it consists of three different components, each capturing a different set of characteristics of posts. We performed a set of experiments to evaluate the effect of each component on summarization. More specifically, we created three instances of MGraph by setting $S_{att} = 1$, $S_{sig} = 1$ and $S_{spec} = 1$ respectively. We also study the effect of not using the topic significance of Equation 7 in the weighting scheme.

The results of Table 11 reveal that the exclusion of any component of the weighting scheme affects negatively the resulting summary. The most important component on McMinn’s dataset [22] is the social attention of posts. Topic significance seems to have limited im-

Table 11: Impact of different components of the weighting scheme. Bold values indicate the highest performing method for the given metric.

McMinn’s Dataset, $N = 10$				
Method	P@N	MRR	α -nDCG@N	AVS@N
MGraph	0.544	0.728	0.774	0.021
$S_{att} = 1$	0.478	0.712	0.699	0.021
$S_{cov} = 1$	0.490	0.701	0.743	0.021
$S_{spec} = 1$	0.512	0.703	0.759	0.017
$D_i = 1$	0.544	0.722	0.721	0.021

Baltimores Riots Dataset, $N = 100$				
Method	P@N	RR	α -nDCG@N	AVS@N
MGraph	1.000	1.000	0.882	0.061
$S_{att} = 1$	0.880	0.500	0.807	0.072
$S_{cov} = 1$	0.910	1.000	0.871	0.068
$S_{spec} = 1$	0.920	1.000	0.818	0.060
$D_i = 1$	0.830	1.000	0.712	0.052

US Elections Dataset, $N = 100$				
Method	P@N	RR	α -nDCG@N	AVS@N
MGraph	1.000	1.000	0.801	0.114
$S_{att} = 1$	0.880	1.000	0.727	0.120
$S_{cov} = 1$	0.760	1.000	0.713	0.145
$S_{spec} = 1$	0.890	1.000	0.692	0.102
$D_i = 1$	0.790	1.000	0.802	0.098

pact as the size of events is small, so the variation in density among topics is also limited. In contrast, this component is one of the most important ones in the other two datasets. The main reason for this is the large number and the diversity of posts in these datasets. In other words, as the structure of the resulting multi-graph G_M becomes increasingly complex, the detected sub-graphs (topics) tend to exhibit wider variation in density. The same observation can be done for the impact of topic coverage. This part seems to be more important on large-scale events with a lot of sub-topics, messages and interactions. On the other hand, social attention tends to be more significant in smaller events.

4.3.7 Impact of topic detection

To study the role of the SCAN algorithm as a topic detection method, we also tested two instances of MGraph using different topic detection techniques. In particular, we used two established probabilistic techniques, namely LDA [5] and TwitterLDA [36]. The rest of the framework components remained the same. In case of LDA, we associate a post to the topic with the higher estimated probability, under the constraint that this probability exceeds a threshold (0.5). The unassociated posts were assigned to single-item clusters in a similar manner as with SCAN. TwitterLDA is an extension of LDA that gives better results for short messages compared to it. The main difference is that TwitterLDA estimates a distribution of topics over users, instead of

Table 12: Comparison of SCAN, LDA and TwitterLDA, used as the underlying topic detection technique in MGraph framework.

Baltimore Riots dataset, $N = 100$				
Method	P@N	RR	nDCG@N	AVS@N
SCAN	1.000	1.000	0.882	0.061
LDA	0.710	1.000	0.801	0.073
TwitterLDA	0.970	1.000	0.837	0.121

US Elections dataset, $N = 100$				
Method	P@N	RR	nDCG@N	AVS@N
SCAN	1.000	1.000	0.801	0.114
LDA	0.790	1.000	0.698	0.121
TwitterLDA	0.920	1.000	0.780	0.191

over posts. Regarding the association of posts to topics, TwitterLDA makes the assumption that each post is associated to a single topic only. This assumption seems to be valid for microblogging posts due to their short length. As these methods require the number of topics K to be defined we set $K = \sqrt{|M|}$, where $|M|$ is the number of posts.

As the size of events (measured by number of posts) in McMinn’s dataset [22] is small, probabilistic models such as LDA have poor performance. We confirmed this hypothesis by inspecting a sample of the resulting topics and found that most of them were not meaningful. This is due to the fact that there are not enough posts to perform a reliable estimation of distributions of topics and words. For that reason, we evaluated SCAN, LDA and TwitterLDA on the other two datasets that consist of hundreds of thousands of posts. The number of topics was set to $K = 462$ and $K = 663$ for Baltimore Riots and US Elections respectively. Considering the single-item topics due to unassigned posts in LDA, the final number of topics rose to 17,079 and 57,639 respectively. In case of SCAN, the number of detected topics (subgraphs) was 2,158 and 2,950 respectively and with the single-item clusters, these rose to 168,974 and 384,437. SCAN detects a much larger number of clusters than the number of topics estimated using the heuristic $K = \sqrt{|M|}$. However, through inspection of a number of single-item clusters, we observed that SCAN left unclustered a large set of posts, the majority of which are outliers. On the other hand, probabilistic topic models created large topics associating a lot of low-quality posts with them, which affected performance in a negative manner.

Table 12 depicts precision and diversity-oriented metrics for MGraph and the two instances using LDA and TwitterLDA. The original version of MGraph achieves the best results. However, comparing these results with the results of Tables 5, 7, 6 and 8, we note that LDA and TwitterLDA still manage to outperform compet-

ing methods, which indicates that even a lower-quality topic detection approach does not have a significant penalty on the performance of **MGraph**. It is also noteworthy that although LDA exhibits significant decrease in most of the metrics, this does not apply for TwitterLDA, which leads to only slightly worse results than the SCAN-based instance of **MGraph**. This is reasonable as LDA is ineffective for short posts, while TwitterLDA was designed with that specific aspect in mind. In addition, the need to manually specify the number of topics for such methods, affects summarization performance in a negative manner as it is highly likely that a sub-optimal choice will be made. This would lead either to multi-topic clusters of posts or to fragmentation of single topics into multiple clusters. The first case would affect precision and recall, while the later would mostly affect diversity-oriented metrics.

Table 13 depicts the top 10 images ranked by **MGraph** for the Baltimore Riots dataset. Note that all the images were labelled as relevant from the annotators. Additionally, each of these 10 images is not a single image but a clique of duplicate images. The promotion of cliques in the first places of the summaries is done mainly for two reasons: First, a clique usually has a higher value of popularity than single images as its popularity is calculated by the sum of individual popularities. Second, cliques are more connected than single images in the posts graph, therefore DivRank gives them a higher score. As these images have a high value of popularity, competing popularity-based methods manage to rank them in high positions as well. For example images in positions 2 and 6 are quite popular, therefore in the first positions of **MostPop** and **P-TWR**. Furthermore, some of these images are part of large clusters corresponding to significant sub-events and discussions during the events (e.g. images 4, 7, 9, 10). For example the image at the 10-th position is related to a discussion about looting stores during the protest. As these clusters are quite large in size, many of these images are also ranked in high positions by topic-based methods. However, big clusters with insignificant images could also benefit from large cluster size (which would introduce noise to the summary). In contrast, **MGraph** manages to combine popularity and topic-based features in a single score and create visual summaries that consist of popular images that at the same time cover multiple sub-topics.

5 Conclusion

We presented **MGraph**, a framework for the generation of visual summaries from social posts related to public events. The key distinguishing characteristic of





MGraph is that it assigns a significance score on each image of the event-related posts, that maximizes the coverage of the underlying topics and the diversity at the same time. We performed a comprehensive experimental study of the method comparing it against a number of state-of-the-art summarization methods on three user-annotated datasets, and concluded that it considerably outperforms existing methods in terms of summary precision (i.e. including relevant images in the summary), while retaining diversity performance at similar levels or even improving it. We also carefully examined the impact of different components of **MGraph** on its overall summarization performance.

In the future, we plan to extend **MGraph** by using more advanced topic detection methods that identify not only topics but also hierarchies and relations between them. Regarding ranking, we plan to investigate the use of co-ranking algorithms to simultaneously rank text and image nodes. Finally, we intend to integrate additional features such as users' popularity, influence and trustworthiness, as recent research indicates that these could improve the results, and especially the quality of selected posts [34].

References

1. Celebrating #SB48 on Twitter. <https://blog.twitter.com/2014/celebrating-sb48-on-twitter> (2014). [Online; accessed 27-Feb-2014]
2. Aiello, L.M., Petkos, G., Martín, C.J., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I., Jaimes, A.: Sensing trending topics in twitter. *IEEE Transactions on Multimedia* **15**(6), 1268–1282 (2013)
3. Alonso, O., Shiells, K.: Timelines as summaries of popular scheduled events. In: Proceedings of the 22nd International Conference on World Wide Web (WWW) companion, pp. 1037–1044 (2013)
4. Bian, J., Yang, Y., Chua, T.S.: Multimedia summarization for trending topics in microblogs. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13, pp. 1807–1812. ACM, New York, NY, USA (2013)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *the Journal of Machine Learning Research* **3**, 993–1022 (2003)
6. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 335–336. ACM (1998)
7. Chakrabarti, D., Punera, K.: Event summarization using tweets. In: Proceedings of 6th AAAI International Conference on Weblogs and Social Media (ICWSM) (2011)
8. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, STOC '02, pp. 380–388. ACM, New York, NY, USA (2002)
9. Chua, F.C.T., Asur, S.: Automatic summarization of events from social media. In: Proceedings of 8th AAAI

Table 13: Summary Example for Baltimore Riots (N=10)

Rank	Image	Text	Rank	Image	Text
1		#BREAKING: MASSIVE BUILDING FIRE AT CHESTER & GAY ST. #BaltimoreRiots @MiriW-BAL	6		Another image you won't see on CNN.
2		Gov. Hogan declares state of emergency in Maryland as protests grow violent. http://cnn.it/1GBqMGI #BaltimoreRiots	7		Who's really behind the #BaltimoreRiots & #BaltimoreUprising? LAdowd and I talk about it https://soundcloud.com/politainment-net/the-right-angle-episode-3
3		Citizens lining up to protect the police on W. North Ave.	8		I really wish y'all would see the bigger picture #BaltimoreRiots
4		Ok Twitter, now lets share pictures of the good side of Baltimore residents #BaltimoreRiots	9		"It was like World War 3" Son describes beatdown from #ToyaGraham during #BaltimoreRiots http://cnn.it/1DDAhx0
5		"I'll just throw this chair through a sports bar and social justice will be restored." #BaltimoreRiots	10		Sneaker stores weren't safe during the #BaltimoreRiots http://trib.al/QqFdFQH

- International Conference on Weblogs and Social Media (ICWSM) (2013)
- Clarke, C.L., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 659–666. ACM (2008)
 - Corney, D., Goker, A., Martin, C., Papadopoulos, S., Mantziou, E., Spyromitros-Xioufis, E., Schinas, M., Iliakopoulou, K., Mironidis, T., Tsampoulatidis, Y., Kompatsiaris, Y., Mass, Y., Aiello, L.M.: D4.3: Social media indexing, aggregation and retrieval. Tech. rep., SocialSensor (2013). URL <http://socialsensor.eu/images/D4.3.pdf>
 - Dork, M., Gruen, D., Williamson, C., Carpendale, S.: A visual backchannel for large-scale events. IEEE Transactions on Visualization and Computer Graphics **16**(6), 1129–1138 (2010)
 - Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research **22**(1), 457–479 (2004)
 - Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: Advances in Neural Information Processing Systems 22, pp. 522–530 (2009)
 - Guille, A., Favre, C.: Mention-anomaly-based event detection and tracking in twitter. In: Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, pp. 375–382. IEEE (2014)
 - Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(1), 117–128 (2011)
 - Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE Transactions on Pat-

- tern Analysis and Machine Intelligence **34**(9), 1704–1716 (2012)
18. Lin, C., Lin, C., Li, J., Wang, D., Chen, Y., Li, T.: Generating event storylines from microblogs. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, pp. 175–184. ACM, New York, NY, USA (2012)
 19. Lu, Y., Zhang, P., Cao, Y., Hu, Y., Guo, L.: On the frequency distribution of retweets. *Procedia Computer Science* **31**, 747–753 (2014)
 20. Mantziou, E., Papadopoulos, S., Kompatsiaris, Y.: Learning to detect concepts with approximate laplacian eigenmaps in large-scale and online settings. *IJMIR* **4**(2), 95–111 (2015)
 21. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: TwitInfo: Aggregating and visualizing microblogs for event exploration. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pp. 227–236. ACM, New York, NY, USA (2011)
 22. McMinn, A.J., Moshfeghi, Y., Jose, J.M.: Building a large-scale corpus for evaluating event detection on twitter. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge management, pp. 409–418. ACM (2013)
 23. McParlane, P.J., McMinn, A.J., Jose, J.M.: Picture the scene...: Visually summarising social media events. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, pp. 1459–1468. ACM (2014)
 24. Mei, Q., Guo, J., Radev, D.: Divrank: The interplay of prestige and diversity in information networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, pp. 1009–1018. ACM, New York, NY, USA (2010)
 25. Nichols, J., Mahmud, J., Drews, C.: Summarizing sporting events using twitter. In: Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces, IUI '12, pp. 189–198. ACM, New York, NY, USA (2012)
 26. Page, L., Brin, S., Motwani, R., Winograd, T.: The page-rank citation ranking: bringing order to the web. (1999)
 27. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005)
 28. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. *Information Processing and Management* **40**(6), 919–938 (2004)
 29. Schinas, M., Papadopoulos, S., Kompatsiaris, Y., Mitkas, P.A.: Visual event summarization on social media using topic modelling and graph-based ranking algorithms. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 203–210. ACM (2015)
 30. Shen, C., Liu, F., Weng, F., Li, T.: A participant-based approach for event summarization using twitter streams. In: Proceedings of NAACL-HLT, pp. 1152–1162 (2013)
 31. Spyromitros-Xioufis, E., Papadopoulos, S., Kompatsiaris, I.Y., Tsoumakas, G., Vlahavas, I.: A comprehensive study over VLAD and Product Quantization in large-scale image retrieval. *IEEE Transactions on Multimedia* **16**(6), 1713–1728 (2014)
 32. Wang, D., Li, T., Ogihara, M.: Generating pictorial storylines via minimum-weight connected dominating set approximation in multi-view graphs. In: AAAI. Citeseer (2012)
 33. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: Scan: A structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07, pp. 824–833. ACM, New York, NY, USA (2007)
 34. Yajuan, D., Zhimin, C., Furu, W., Ming, Z., Shum, H.Y.: Twitter topic summarization by ranking tweets using social influence and content quality. In: Proceedings of the 24th International Conference on Computational Linguistics, pp. 763–780 (2012)
 35. Yang, Z., Guo, J., Cai, K., Tang, J., Li, J., Zhang, L., Su, Z.: Understanding retweeting behaviors in social networks. In: Proceedings of the 19th ACM international conference on Information and knowledge management, pp. 1633–1636. ACM (2010)
 36. Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing twitter and traditional media using topic models. In: Advances in Information Retrieval, pp. 338–349. Springer (2011)