Multimodal Graph-based Event Detection and Summarization in Social Media Streams

Manos Schinas
CERTH-ITI
Thessaloniki, Greece
manosetro@iti.gr

Symeon Papadopoulos
CERTH-ITI
Thessaloniki, Greece
papadop@iti.gr

Georgios Petkos
CERTH-ITI
Thessaloniki, Greece
gpetkos@iti.gr

Yiannis Kompatsiaris
CERTH-ITI
Thessaloniki, Greece
ikom@iti.gr

Pericles A. Mitkas
Aristotle University of
Thessaloniki, Greece
mitkas@eng.auth.gr

ABSTRACT

The paper describes a multimodal graph-based system for addressing the Yahoo-Flickr Event Summarization Challenge of ACM Multimedia 2015. The objective is to automatically uncover structure within a collection of 100 million photos/videos in the form of detecting and identifying events, and summarizing them succinctly for consumer consumption. The presented system uses a sliding window over the stream of multimedia items to build and maintain a multimodal same-event image graph and applies a graph clustering algorithm to detect events. In addition, it makes use of a graph-based diversity-oriented ranking approach and a versatile event retrieval mechanism to access summarized instances of the events of interest. A demo of the system is online at `http://mklab.iti.gr/acmmm2015-gc/`.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Experimentation, Algorithms

Keywords

Event Detection, Clustering, Event Summarization

1. INTRODUCTION

As a result of the widespread use of media capturing devices and social media sharing platforms, a growing amount of diverse multimedia content is available online. Hence, there is a profound need for information systems that can effectively organize such content. As real-world events are a key part of social life, the detection of events and the organization of content around them is an effective way for navigating and searching large multimedia collections.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
MM'15, October 26–30, 2015, Brisbane, Australia.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3459-4/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2733373.2809933 .

Event detection in social media is a challenging task due to the nature and characteristics of social multimedia: There is a large number of multimedia items that are of personal nature and/or not relevant with respect to an event. Also, many items have missing fields, e.g., title or taken date, that make difficult their linking to an already known event. Additionally, the order of magnitude of available media content creates excessive computational costs and calls for scalable and efficient solutions. Furthermore, the presentation of detected events poses additional challenges as the volume of images for large scale events e.g., the Olympic Games, could be massive. Not every image is of the same importance for an event and long-running events can be quite complex containing many aspects and numerous sub-events. Thus, there is a need for event-based summarization methods that can produce concise visual summaries for any time interval of the event, covering its main aspects.

In this paper, we describe a prototype system addressing the Yahoo-Flickr Event Summarization Challenge of ACM Multimedia 2015. We detect events from a set of about 100 million images [12] and then summarize each detected event by presenting only a representative and diverse subset of images. Although the system uses Flickr images, it could be applied to any multimedia collection with a similar set of metadata. To handle the massive scale of the dataset, the system uses a sliding time-window that enables the discovery of new events or their merging into previously detected events. During the procedure, an image graph is constructed and updated on the fly, and a graph-based clustering algorithm is applied periodically to detect dense subgraphs that correspond to events. In order to keep track of evolving or periodic events, a graph of events is also generated incrementally. To perform summarization, the system takes advantage of the image graph and applies per event a graph-based ranking algorithm that produces a diverse set of images, ranked by their importance.

2. RELATED WORK

Numerous approaches have recently appeared in the literature on the problems of event detection tracking and summarization in social media content. There are two general event detection and tracking approaches: *document-pivot* [1, 9] and *feature-pivot methods* [4]. The former aim to cluster items related to the same event. The latter aim to first iden-

tify the representative features (e.g., tags) of the underlying events in the collection (typically by detecting bursty frequency patterns along time or space) and then detect events by leveraging these representative features.

The core idea of the majority of works in event summarization is the segmentation of items set into coherent topics or sub-events and the selection of the most representative documents in each segment. For event segmentation several approaches have been proposed, ranging from Hidden Markov Models [3] to LDA extensions [2]. Regarding selection of representative documents, either centroid-based [10] or graph-based ranking approaches [5] have been considered.

3. SYSTEM OVERVIEW

The proposed system consists of three main components: event detection and tracking, event summarization, and event retrieval and presentation. The event detection and tracking component identifies events per timeslot and keeps track of their evolution from timeslot to timeslot. The summarization component calculates a score for each image per event by applying a graph-based ranking algorithm. Finally, event retrieval and presentation is the online component of the system, used to present the detected events to end users.

3.1 Event Detection and Tracking

The system utilizes a sliding time-window to detect and keep track of events. As the window moves along the timeline, new images are inserted into the window while others are removed. As days are a meaningful unit of time for events, we set the step of the time-window to one day and its length to three days. In other words, there is a significant overlap of two days between successive timeslots. We use this overlap to keep track of evolving and long-running events as will be described in the next sections.

For images within a given timeslot we use what is termed the Same Event Model (SEM) [7]. The SEM takes as input the set of per modality similarities between two images and predicts how likely it is that these two images belong to the same event. A graph of images $\mathcal{G}_I$ is constructed, in which the existence of an edge between a pair of images denotes the positive prediction of the SEM. Subsequently, a graph-based clustering algorithm is applied on $\mathcal{G}_I$ to obtain a full clustering. Finally, the method either merges these clusters into events from previous timeslots or creates new events.

3.1.1 Same Event Model

To implement the SEM, a Support Vector Machine (SVM) with a linear kernel is utilized. An independent training dataset from the 2014 MediaEval Social Event Detection task (SED 2014) [8] was used for building the model. The SED 2014 dataset contains 362,578 Flickr images that are associated with 17,834 events. Several variations were considered for building the models, including: a) different ways of sampling the training and test data, b) different sets of features and feature similarities, and c) different values of the classification threshold.

According to [9], sampling of the training and testing data was found to be a critical factor for the usefulness of the learned SEM. The problem arises because in reality, there are pairs of images that have some similarity, according to at least some modalities, e.g., time, but do not actually represent the same event. However, it is rather unlikely that random sampling of negative instances, picks pairs of items that are even marginally similar to each other. An alternative is to sample negative pairs so that the chosen pairs of items are similar according to some modality. Empirically, we observed that by doing this, although training accuracy dropped test accuracy increased because the model was more effective on difficult (borderline) cases compared to a model that would have been trained on purely randomly selected data. To be more specific, the models that were eventually used were trained with 80% randomly selected data and 20% non-randomly selected data. Each data point that was not randomly selected involved the examination of a single modality at a time, and all used modalities were equally considered for sampling the 20% of non-randomly selected data. The single-modality similarities were based on the following features:

- **Taken time:** Four similarity scores are extracted. The first is the absolute difference in days between the two image timestamps. The other three are binary indicators signifying whether the difference between the time each of the images was taken is smaller than 3 hours, 12 hours and 24 hours respectively.
- **Text:** Three text metadata fields are used, namely title, description, and tags. For each field, we compute a $tf \cdot idf$ vector and the corresponding cosine similarity. Additionally, we extract named entities from these fields by using Stanford NLP library[1] and include three additional similarity measures based on the number of common locations, organizations and names of people.
- **Image:** We used SURF+VLAD using the implementation of [13]. In particular, we extract SURF features from the raw image content and aggregated them using the VLAD scheme. We then used the L_2 distance between VLAD vectors to compute similarities.

The third variation that we explored for building the SEMs was to tune the classification threshold. Intuitively, for each image, the SEM will be evaluated against far more images than the average event size. In a typical scenario, the "real negative" evaluations of the SEM are expected to be much more than the "real positive" ones. Therefore, in order to produce a "clean" graph with not too many spurious edges, it is useful to increase the true negatives rate. This can be achieved by increasing the classification threshold for positive predictions, at the cost of a lower true positives rate. According to our experiments, this significantly improves the clustering performance. The test accuracy of the model when the threshold is not tuned is 98.65% for positives and 98.81% for negatives (average 98.73%). When the threshold for positive classification is set to 0.9, test accuracy for positives drops to 95.12%, while accuracy on negatives increases to 99.87% (average accuracy is 97.49%).

3.1.2 Generation of Image Graph

As the time-window moves forward, new images are inserted into the image graph $\mathcal{G}_I$. At the same time, the out-of-window images and the corresponding edges are removed from the graph. For the newly inserted images, edges between them and images already in the graph are calculated based on the SEM as described in section 3.1.1. To limit the number of SEM evaluations and make the approach more

[1]http://nlp.stanford.edu/software/CRF-NER.shtml

scalable, we apply a candidate neighbor selection step [9]. For each image, we utilize appropriate indices to obtain the most similar images according to each modality and evaluate the SEM only for them. More specifically, for each image we retrieve all the nearest images in terms of textual and visual content provided that they exceed a predefined threshold and are within the current timeslot. Then, we calculate the same event score only for the union of these two sets. For text we used an indexing scheme based on Locality Sensitive Hashing, while for visual content, we used Product Quantization on top of the SURF+VLAD representation.

To detect the events in a time window we opted for a graph clustering algorithm, namely the Structural Clustering Algorithm for Networks (SCAN) [14]. We apply SCAN on $\mathcal{G}_I$ to identify dense sub-graphs of images. These sub-graphs represent the events in the set of images of this time window, i.e. each event $e \in E$ is represented as a set $\mathcal{G}_e$ of highly connected images on the graph. However, a substantial amount of images is kept outside of the detected clusters. These images are divided into two categories, *hubs* and *outliers*. Hubs are bridges to more than one clusters, while outliers are images that may be connected to images of a cluster but do not belong to any of them. We add hubs to the clusters to which they are connected under the condition that the number of edges exceed a predefined threshold (10 edges). Note that after this step hubs may belong to more than one clusters. Once the set of events E are detected, we use images associated with each $event_i \in E$ to calculate an aggregated representation of the event. Namely, for each event, we compute the minimum, maximum and average taken time of the images and calculate three merged $tf \cdot idf$ vectors that describe the aggregated textual content of title, description and tags. As named entities are expected to be particularly relevant for each event, we also aggregate persons, locations and organizations.

3.1.3 Event Tracking

As new events are detected, the system tries to link them with detected events from earlier timeslots. To identify that two clusters are referring to the same event we use the structural similarity between the underlying sub-graphs, expressed as the Jaccard coefficient of their edges. If this similarity exceeds a certain threshold (0.6), we merge the two clusters into a single event. Intuitively in our case, if two sub-graphs of successive timeslots share a significant amount of edges, the later sub-graph is arising from the previous one. Note that the overlap of edges is feasible due to the content overlap (of two days) between successive timeslots. When such a similarity is detected, the new event is merged into the older. Namely, the images of the new event are added to the previous event and its aggregated representation is updated accordingly. Furthermore, we also check for *inactive* events. As inactive, we define the events that have no new inserted images for the last three days. As the length of the timeslot is also three days, it is unlikely that these events will have any overlap with events in the following windows. We remove those from the set of active events and discard events with less than 10 images. For the rest, we calculate whether they are *local*, *regional* or *global*. More specifically, for each event we find a median point and an average radius based on the latitude and longitude of its images. Events with radius <100km are considered to be local. Between 100 and 500km they are classified as regional, while events with

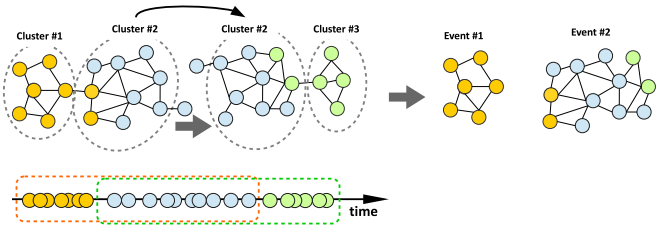

Figure 1: Event tracking in successive timeslots.

average radius >500km are defined as global. Also, events with images from a single user are tagged as *personal*. Finally, we calculate the significance scores of images as will be described in the next section, and we store and index them in MongoDB and Solr respectively.

3.2 Event Summarization

Our goal is to generate a concise summary of a selected event e for an arbitrary time-frame. A summary is a set of representative images depicting the key aspects of the selected time interval. To support this at retrieval time, we calculate a significance score for each event image according to its position on the sub-graph $\mathcal{G}_e$. This calculation is based on the *MGraph* summarization method [11], which first finds sub-events within the event and uses these sub-events to calculate a prior score for each image. Then, to incorporate diversity into image scoring, it uses DivRank [6], a variant of PageRank that aims at diversity. In our case, instead of finding sub-events within an event by clustering its images, we perform time-based segmentation. Namely, we create segments/clusters corresponding to one-day timeslots, and apply scoring in the same way as done in [11]. At the end of this process, for an event lasting D days, we have a rank of the images within each time-slot $t_i, i = 1...D$.

3.3 Event Retrieval and Presentation

At retrieval time, there are two main tasks: The first is to obtain a set of events that match a specific query q. The second is to use the calculated DivRank scores to generate a summary of arbitrary length.

Regarding retrieval, the initial query q is usually short, e.g., `olympic games`, and although this usually yields results of high relevance, it cannot fully capture the underlying information need of the user. To improve the expressive power of q and increase recall, query expansion is applied: $q \rightarrow q_{ext}$. For this purpose we use the tags and named entities of the retrieved events to expand the query with new terms. As single tags and named entities could yield noisy results, we combine them in pairs. The final query is the logical OR between the aforementioned pairs.

For each retrieved event e with associated images I, the user can select to see a summary of length L for a specific time-frame $\mathcal{F}$. To this end, we use the scores described in section 3.2. To achieve this, we keep images $I_{\mathcal{F}}$ within $\mathcal{F}$ and iterate over the timeslots of $\mathcal{F}$ to form a summary following a greedy approach. We first calculate the compression rate $CR = L/|I_F|$, needed to achieve a summary of length L. Then, for each timeslot $t_j \in \mathcal{F}$, we get the corresponding images I_j and find the number of images that need to be selected to meet the budget of L messages. Namely, the local target for timeslot t_j , is $L_j = CR \cdot |I_j|$ messages.

Table 1: Details of predefined events

Event	#events	#items	i/ev	u/ev
Occupy Movement	120	9,814	81	3.45
Batkid	2	624	312	2
Olympic Games	117	11,535	98	5.2
Eyjafjallajokull Eruption	12	588	49	1.7
Holi, festival of colors	2	52	26	1
Byron Bay Bluesfest	9	707	78	2.4
Hanami	16	3106	194	6.6

Table 2: Basic Statistics of event detection process between 1/1/2009 and 1/1/2014

Statistics	Personal	Social	Total
Events	23,184	45,729	68,913
Avg. items/event	39	47	44
Avg. users/event	1	5.25	3.82
Local	17,334	24,453	41,787
Regional	765	11,196	11,961
Global	171	7,929	8,100
Avg duration (hours)	19.11	42.9	34.84
Max duration (days)	9.83	19.25	19.25

The algorithm iterates through the images in I_j sorted by their DivRank score and selects the top L_j images provided that the value of Equation 1 is below a predefined threshold R_{th}. As the maximum possible L_2 distance is $\sqrt{2}$ we set $R_{th} = \sqrt{2}/2 \simeq 0.7$. In case that the length of a local summary is still below than the desired one, we re-calculate the compression rate to select more images from the next timeslots.

$$Redundancy(im) = \min_{im' \in S} L_2(im, im') \quad (1)$$

The score of Equation 1 measures the redundancy introduced by an image im with respect to the set of already selected images. To this end, we compute its minimum visual distance, based on L_2, to the already selected images. Filtering based on redundancy during image selection ensures that no visual duplicates are including in the summary.

4. RESULTS

Table 1 presents some statistics for the set of seven predefined events. The number of sub-events in each event and the number of associated items exhibits high variance. Large scale real-world events, such as the `Occupy Movement` and the `Olympic Games`, consist of several sub-events and the number of items per event is high. Also, the average number of users per event is higher as more users are engaged in these events. On the other hand, single day events, such as the `batkid` case, consist of a small number of sub-events with many associated items.

Table 2 presents some basic statistics for all the events detected in the YFCC dataset between 1/1/2009 and 1/1/2014. In total, there are 68,913 events, of which 45,729 are social events, i.e. the multimedia content for the same event comes from more than two users. Regarding the characteristics of personal and social events, a marked difference is the following: personal events are typically tagged as local, since it is unlikely that the same user can post content for the same event from different locations. At the same time, the ratio of events tagged as regional and global is considerably higher for social events.

5. ACKNOWLEDGMENTS

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

6. REFERENCES

[1] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 291–300, New York, NY, USA, 2010. ACM.

[2] J. Bian, Y. Yang, and T.-S. Chua. Multimedia summarization for trending topics in microblogs. In *Proceedings of the 22nd ACM international conference on Conference on Information and Knowledge Management*, CIKM '13, pages 1807–1812, NY, USA, 2013. ACM.

[3] D. Chakrabarti and K. Punera. Event summarization using tweets. *ICWSM*, 11:66–73, 2011.

[4] L. Chen and A. Roy. Event detection from flickr data through wavelet-based spatial analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 523–532, New York, NY, USA, 2009. ACM.

[5] 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.

[6] Q. Mei, J. Guo, and D. Radev. 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, pages 1009–1018, NY, USA, 2010. ACM.

[7] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris. Social event detection using multimodal clustering and integrating supervisory signals. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, ICMR '12, pages 23:1–23:8, New York, NY, USA, 2012. ACM.

[8] G. Petkos, S. Papadopoulos, V. Mezaris, and Y. Kompatsiaris. Social event detection at mediaeval 2014: Challenges, datasets, and evaluation. In *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain, October 16-17*, 2014.

[9] G. Petkos, S. Papadopoulos, E. Schinas, and Y. Kompatsiaris. Graph-based multimodal clustering for social event detection in large collections of images. In *Proceedings of the 20th Anniversary International Conference on MultiMedia Modeling - Volume 8325*, MMM 2014, pages 146–158, New York, NY, USA, 2014. Springer-Verlag New York, Inc.

[10] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[11] M. Schinas, S. Papadopoulos, Y. Kompatsiaris, and P. Mitkas A. Visual event summarization on social media using topic modelling and graph-based ranking algorithms. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, ICMR 2015, 2015.

[12] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[13] E. S. Xioufis, S. Papadopoulos, Y. Kompatsiaris, G. Tsoumakas, and I. P. Vlahavas. A comprehensive study over VLAD and product quantization in large-scale image retrieval. *IEEE Transactions on Multimedia*, 16(6):1713–1728, 2014.

[14] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: A structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 824–833, New York, NY, USA, 2007. ACM.