News-oriented multimedia search over multiple social networks

Article · July 2015

DOI: 10.1109/CBMI.2015.7153612

3 authors:

Katerina Iliakopoulou
The New York Times
4 PUBLICATIONS **130** 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

News-oriented multimedia search over multiple social networks

Katerina Iliakopoulou, Symeon Papadopoulos, Yiannis Kompatsiaris
Information Technologies Institute (ITI)
Centre for Research and Technology Hellas (CERTH)
Thessaloniki, Greece
Email: {ailiakop,papadop,ikom}@iti.gr

***Abstract*—The paper explores the problem of focused multimedia search over multiple social media sharing platforms such as Twitter and Facebook. A multi-step multimedia retrieval framework is presented that collects relevant and diverse multimedia content from multiple social media sources given an input news story or event of interest. The framework utilizes a novel query formulation method in combination with relevance prediction. The query formulation method relies on the construction of a graph of keywords for generating refined queries about the event/news story of interest based on the results of a first-step high precision query. Relevance prediction is based on supervised learning using 12 features computed from the content (text, visual) and social context (popularity, publication time) of posted items. A study is carried out on 20 real-world events and breaking news stories, using six social sources as input, and demonstrating the effectiveness of the proposed framework to collect and aggregate relevant high-quality media content from multiple social sources.**

I. Introduction

During the last decade, the rise of Online Social Networks (OSNs) has revolutionized how news stories are distributed and events are covered. In recent years, social media are becoming an increasingly popular means of exchanging information, and their extensive use generates huge amounts of data, both text and multimedia, which is shared when a news story breaks or an event takes place [1]. This direct means of broadcasting has contributed to reconsidering the practices of conducting journalism, by increasingly exploiting the abundance of information communicated in social networks for detecting news stories and monitoring events [2].

In this context, Twitter has grown to be a significant news source [3], [4] and has been extensively studied as a means of monitoring and collecting data around breaking news [5], [6]. Apart from Twitter, other social networks, such as Facebook, Google+, Instagram, Tumblr and Flickr, have emerged as increasingly important channels of multimedia content around events and news stories. Yet, despite the fact that the same news story or event is covered in different, complementary to each other, ways depending on the OSN platform, there is currently no straightforward and effective means of searching for news-focused multimedia over multiple OSN sources.

In order to collect photos and videos posted by users in social networks, it is necessary to build appropriate queries related to the event or story at hand, and to use them for performing requests to the respective APIs. Although finding hashtags and keywords relevant to an event is in some cases straightforward, extracting the essence of a story to build a representative query is often challenging. There is the risk of long complicated queries that retrieve no results, as well as of rather vague queries bringing back irrelevant content. Some OSNs, such as Flickr, might be more flexible, being able to return numerous results that contain all requested keywords or a portion of them with the appropriate ranking, whereas others, such as Instagram, can handle only hashtags and consequently return zero or very few results when they are given many keywords as input. Another crucial aspect is the order of keywords in the query. In some OSNs, the keyword order makes no difference, whereas in others, different keyword order might bring back a totally new set of results. Consequently, a query might easily fail in case it is not built according to the requirements of the respective OSN.

978-1-4673-6870-4/15/$31.00

When assessing multimedia content in relation to an event or news story, the following properties are crucial: a) high relevance to the topic of interest, b) high quality of multimedia, c) diversity of the retrieved media, d) usefulness with respect to usage for reporting and publication purposes. Although recent research has focused on the optimization of query formulation methods utilizing terms, proximities and phrases with respect to their frequency and text position [7], [8], [9], [10], and also through modelling query concepts [11], [12], there has been no work targeted at the problem of retrieval over multiple OSN platforms. To achieve the above goals and overcome the challenges of multi-OSN search, the paper presents and evaluates a novel multimedia retrieval framework, making the following contributions:

A novel graph-based query formulation method, catered for the special traits of each OSN, that captures the primary entities and topics of the event of interest and their associations, builds a large set of queries by a greedy graph traversal algorithm, and ranks them by relevance and diversity.

A relevance classification method that computes 12 features from a set of search results based on their content (text, visual) and context (popularity, publication time). The features are used in a supervised learning manner for ranking the collected images based on the requirements described above.

A real-world evaluation of the proposed framework on a set of 20 events and news stories involving a total of more than 88K images. Through human assessment of the retrieved results, we demonstrate the effectiveness of our framework.

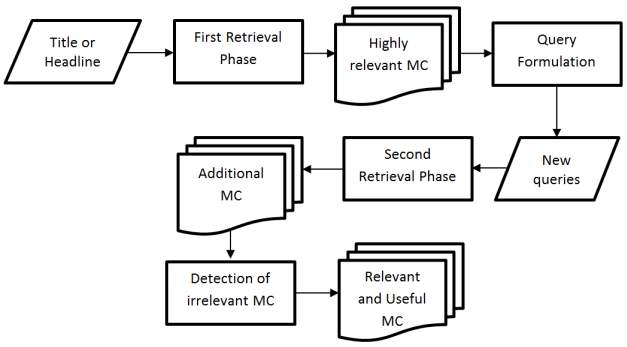


Fig. 1: Overview of proposed multiple OSN search approach. MC stands for Multimedia Content.

II. Framework description

The input to the proposed search process is the headline of the news story or title of the event of interest. Given this, the first step involves the collection of a set of media items using a high-precision query to the respective OSN search APIs. The collected results are then used by a graph-based query formulation and ranking method with the goal of producing multiple queries around the news story/event of interest. Those are then submitted to the OSN search APIs and a second set of results are collected. In the final step, we employ a relevance classifier with the goal of discarding irrelevant or low-quality results. Figure 1 gives an overview of the framework, while the next paragraphs detail the depicted steps.

Notation: We will use $M = \{m\}$ to refer to sets of media items. In our experiments, we restrict to images, since the relevance classifier makes use of some features that are image-specific[1]. Subscripts are used to further specify a particular set of media items, e.g., M_0 corresponds to the set collected from the first query, while M_{ext} corresponds to the extended set collected from the second query. Typically, media items retrieved from OSN APIs are accompanied by a number of metadata attributes. We use functions to refer to these, e.g., title(m_i) refers to the title of media item i, while date(m_i) refers to its publication date. We denote queries by q, and sets of queries by $Q = \{q\}$. Also, we denote single keywords by k and hashtags by h, and to weighting functions, such as frequency count as freq(k).

A. Collection of highly relevant content

As a first step, the proposed method builds a collection M_0 of multimedia items that are largely highly relevant to the investigated story/event. For that purpose, we query the six selected social networks with a high-precision query q_0, that in our case corresponds to the news story headline or to the official name of the event. An additional restriction employed to lower the possibility of collecting noisy content, is discarding all material retrieved before the news story broke or the official date the event started, i.e. $M_0 = \{m|\text{date}(m) > t_0\}$, where t_0 is the starting date of the story/event. Even though all OSNs discussed in this study are used in this first step, only three of those, Google+, Flickr and Twitter, were found to contribute to the collection. This is expected given the retrieval behaviour each OSN exhibits.

[1]However, the query building method is independent of the type of medium sought, i.e. it is also applicable to videos.

B. Keyword and hashtag extraction

The goal of this step is to detect keywords and hashtags that are representative of the topic and reveal different aspects of it. We first detect and extract the set of Named Entities (NE) appearing in the text metadata of M_0. We then perform text pre-processing to discard all stopwords and filter out HTML tags, web links and social network account names. In addition, we perform stemming for keywords that are not listed as NEs, to group keywords with similar meaning and investigate their connections to other concepts. In the end, a list of keywords is formed, where each keyword k is associated with a frequency count freq(k). Note that in case the title and the description of a multimedia item match, only one of them is counted to avoid skewing the frequency computation.

We choose to treat keywords and hashtags separately due to the fact that hashtags are placed independently inside text without following or preceding a related hashtag or keyword. Consequently, a separate list of popular hashtags is created, similar to the keywords list described above, where each hashtag is associated with its own weight. This is computed as the sum of its own frequency of appearance and the frequencies of keywords that are part of the hashtag (since it is typical for a hashtag to consist of multiple keywords):

$$\text{weight}(h) = \text{freq}(h) + \sum_{k \subset h} \text{freq}(k) \quad (1)$$

where $k \subset h$ is used to denote that keyword k is part of hashtag h. For example, for the Australia Open Tournament event, the hashtag `#australiaopen` is boosted by the frequency counts of both `australia` and `open`. After this step, we end up with two sets, namely $K = \{(k, \text{freq}(k))\}$ and $H = \{(h, \text{weight}(h))\}$.

C. Keyword graph construction

In this step, we construct a keyword graph $G = (V, E)$ to support the query building process (to be described in subsection II-D). The vertices of the graph correspond to the set of selected keywords, $V = \{k\}$, while the edges represent their pairwise adjacency relations, where adjacency is computed with respect to the text metadata of M_0. For instance, the keyword sequence `k1 k2 k3` in a piece of text would result in increasing the frequency of edges (`k1`, `k2`) and (`k2`, `k3`) by 1. Hence, the graph is directed, i.e. (`australia`, `open`) and (`open`, `australia`) are two different edges. Note that each edge $e \in E$ is associated with a frequency freq(e) that expresses the frequency of appearance of the phrase composed of the edge keywords in the set of results. Since the graph is directed, for each node k, we define its in- and out-degree as $\deg_{in}(k)$ and $\deg_{out}(k)$ respectively.

Only significant keywords about the story/event of interest are considered for the graph construction. This serves both the elimination of noisy keywords and the cost-effectiveness of the method in terms of running time. For that purpose, the average frequency of keywords weights is calculated and only keywords that have greater frequency than the average

are used for the graph creation. Nevertheless, irrelevant, vague or of little importance to the subject keywords may still appear in the constructed graph. Such nodes can be identified due to their low connectivity to the rest of the graph. To get rid of such nodes, we first filter edges with $\text{freq}(e) > \theta^-$ (empirically set to 3), and then retain only nodes with $\deg_{in}(k) > d_{in}^-$ and $\deg_{out} > d_{out}^-$ (both empirically set to 2).

D. Query building

After the graph construction, the framework creates two lists of queries: a keyword-based and a hashtag-based list.

Keyword-based queries: These queries are created via a graph traversal process, since a keyword-based query can be considered as a path from a starting node to an end node on the graph, given a starting node k_0 and a maximum number L of hops. In order for a node to be considered a starting node, it has to possess a high out-degree. Nodes with lower out-degree but connected to heavy weighted edges are also regarded as good starting nodes. A total score for each node is computed which is based on both factors:

$$\text{score}(k) = \deg_{out}(k) + \frac{1}{\deg_{out}(k)} \sum_{k_n \in N_{out}(k)} \text{freq}(k, k_n) \quad (2)$$

where $N_{out}(k)$ denotes the set of nodes that are connected to k with out-going edges. The average score over all graph nodes is computed and only nodes with a score above average are selected as starting nodes, making up the set K_S. L is set equal to the length of the average path on the graph.

Traversing all possible paths on the graph should result in a large set of diverse queries. Beginning at each iteration from one of the selected nodes k_0, the algorithm performs a local expansion process in a recursive way until the maximum number of steps is reached or no other unvisited node is left to be included in the path. At each step, the weight of the edge that connects the newly traversed node with its parent is accumulated resulting in the score assigned to the final query. The latter is normalized after the traversal is complete by the number of steps. At the end, all queries are ranked on the basis of their associated scores. Algorithm 1 provides a precise description of how the algorithm operates.

To avoid query duplication, we also perform a reranking step, in which we penalize queries that exhibit high text similarity, computed by the Jaccard coefficient, to queries with a higher score. This ensures that the top ranking queries will be as diverse as possible, and therefore should capture multiple aspects of the story/event of interest.

Hashtag-based queries: We first adjust the weights of hashtags from the set H (see subsection II-B), using the adjacency weights of G. More specifically, for all keywords embedded in a hashtag, the hashtag weight is boosted by the edge weight that connects the two keywords in the graph:

$$\text{weight}'(h) = \text{weight}(h) + \sum_{k_i, k_j \subset h} \text{freq}(k_i, k_j) \quad (3)$$

This action boosts the rank of multi-keyword hashtags and ensures that generic hashtags are omitted from the final queries.

```
for k_0 ∈ K_S do
    q ← k_0;
    score(q) ← 0;
    v ← k_0;
    l ← 1;
    traverseGraph(v);
end
traverseGraph(v);
for k ∈ N_out(v) do
    if k ⊂ q then
        score(q) ← score(q)/l;
        return q;
    end
    if l = L then
        score(q) ← score(q)/L;
        return q;
    end
    q ← q||k;
    score(q) ← score(q) + weight(v, k);
    traverseGraph(k);
end
```

Algorithm 1: Graph traversal for generating a large set of queries using the graph keywords. v stands for the current node, l for the number of steps performed so far, and $a||b$ the concatenation between keywords a and b (adding a whitespace in between).

At the end of the query building process, we retain the top M keyword-based queries Q_K and the top N hashtag-based queries Q_H for use in the second round of media collection. Since the number of appropriate queries differs among events or news stories, to determine M and N in an unsupervised way, we seek significant gaps between the scores of successive queries, since it is an indication that the quality of the formulated queries starts to drop significantly. To bound the complexity of the subsequent media collection step, we also set maximum allowed values M_{max} and N_{max} for the number of keyword- and hashtag-based queries respectively.

E. Relevance classification

Submitting the queries of $Q_K \cup Q_H$ to the OSN APIs, we end up with an extended collection of multimedia $M_{ext} = M_K \cup M_H$. A significant proportion of the collected multimedia content might be noise or irrelevant, including selfies, sketches, TV-shots and memes. To filter undesired content, we employ a relevance classification step, in which we attempt to predict the relevance and quality of an item $m \in M_{ext}$ based on a number of features that express different aspects of its content and impact. To this end, we consider that a set M_L of labelled images are available at our disposal, consisting of positive (relevant, high-quality) and negative (noise, irrelevant) examples, denoted as M_L^+ and M_L^- respectively.

More specifically, we extract 12 image features that are listed and briefly described in Table I. Five of the features are *popularity*-based. Four of them are computed on the basis of the relevance of the media item in terms of text content. One of them is computed using visual similarity. Another one expresses the temporal proximity of the media item to the story or event. The final feature is computed on the basis of the image dimensions.

Popularity features include the number of likes, views, shares and comments attracted by a published image. We also

consider their sum as an additional feature. No normalization is performed on the values obtained from the respective APIs. Although we recognize that different OSNs exhibit different statistics regarding popularity measures (e.g., the average number of comments on Facebook is quite different compared to the one on Flickr), we still consider their raw values as valuable cues to the target classifier, and we leave this as an open problem for future work.

Text-based relevance features are computed with reference to the initial high precision query q_0 (that was employed to collect the first set of multimedia content as described in subsection II-A). More specifically, for the *title* and *description*, a text similarity score is computed based on the frequency of query keywords in them:

$$\text{match}(q, T) = \frac{1}{\text{len}(T)} \sum_{k \in q} \text{freq}(k, T) \quad (4)$$

where T denotes the title or description field of item m and $\text{freq}(k, T)$ denotes the frequency of appearance of keyword k in T. In case of *tags*, the matching score is computed as:

$$\text{match}(q, K_T) = \frac{|q \cap K_T|}{|K_T|} \quad (5)$$

where K_T denotes the set of tags. In both cases, stop words are removed from the respective text fields. In addition, an aggregate text matching feature is computed based on the three text features by summing them and adjusting the score in case the description and/or the tags fields are missing.

The similarity feature plays an important role in detecting relevant images, as it corresponds to a score expressing the visual similarity between the image in question and any of the positive labelled images M_L^+. The similarity is computed by computing the Euclidean distance between the VLAD+SURF vector (based on the implementation of [13]) of the examined image with the ones extracted from the images of M_L^+ and taking the maximum similarity as the value of the feature.

The date of when the image being examined was published in the social network is also an indicator of its relevance to the story. Usually, images posted soon after the time t_0 the event started/ended or the news story broke are more likely to be of relevance, while this possibility decreases as the time difference grows. We quantize the temporal difference between $\text{date}(m)$ and t_0 into three values corresponding to publication during the event ($<$1 day), short after the event ($<$2 days) and long after the event ($>$2 days).

TABLE I: Image features used for relevance classification.

Feature	Description
Likes	Number of likes
Views	Number of views
Comments	Number of comments
Shares	Number of shares
Popularity	Sum of Likes, Views, Comments and Shares
Title	Textual match of title(m) with q_0
Description	Textual match of desc(m) with q_0
Tags	Textual match of tags(m) with q_0
Total Text	Sum of Title, Description and Tags values
Similarity	Maximum visual similarity to the images of M_L^+
Date	Time difference (in days) between date(m) and t_0
Size	Image size category (small, medium, large)

As last feature we consider one based on the size of the image. We quantize the image size (in terms of number of pixels) into *small*, *medium* and *large*, depending on whether it is smaller than a 200×200 image, between a 200×200 and a 500×500 one, or larger than a 500×500 respectively.

III. Evaluation

The proposed method was tested on the set of 20 events and news stories listed in Table II. Much of the multimedia content that is posted on social networks is deleted after a short period of time, therefore we confined our choice of events and news stories to a strict period of five months just before the data collection took place (November 2013 - March 2014) to ensure the collection of representative and sufficient material for each one. Additional criteria for our selection of events and stories were their size and variety. Events and stories of large size are necessary to test the effectiveness of the proposed approach to build diverse queries, and a varied selection made sure that we covered a representative set of real scenarios.

We set $M_{max} = 20$ and $N_{max} = 10$, setting the maximum number of keyword-based queries higher due to the larger lexical variation that free-form text queries have.

In addition to the list of events, Table II provides the number of images collected from all OSNs during the first (M_0) and second (M_{ext}) steps of the querying process, corresponding to the high precision query and the set of queries $Q_K \cup Q_H$ respectively. In total, a set of more than 88K images were collected for all 20 events and news stories, leading to an average of approximately 4.4K images per event/news story, with events being associated on average with more images (5.5K) compared to news stories (3.3K).

Media volume per OSN: We first examined the amount of content that is retrieved from each social network. Based on our measurements, Flickr contributes the most to the collected media content in both collection steps, with Twitter following in terms of contribution. Instagram and Google+ result in less content compared to Twitter, but still their contribution is measurable. Tumblr and Facebook contribute the least amount

TABLE II: List of test events and news stories.

#	Name	$\lvert M_0 \rvert$	$\lvert M_{ext} \rvert$	×
	Events			
1	Australian Open Tennis 2014	719	3,553	4.9
2	67th British Academy Film Awards	4,367	16,596	3.8
3	Brit Music Awards 2014	1,329	2,248	1.7
4	71st Golden Globe Awards	2,517	3,406	1.4
5	Sundance Film Festival 2014	18	8,592	477
6	56th Grammy Awards	759	2,197	2.9
7	86th Academy Awards	1,297	4,597	3.5
8	Winter Olympic Games 2014	3,384	6,174	1.8
9	Superbowl XLVIII	164	5,498	33.5
10	Victoria's Secret Fashion Show 2013	1,563	2,410	1.5
	News stories			
1	Ariel Sharon's death	70	793	11.3
2	2014 Crimean crisis	58	3,677	63.4
3	Winter storm South US	215	4,631	21.5
4	Malaysian Airlines flight missing	297	3,198	10.8
5	Philip Seymour Hoffman's death	344	4,266	12.4
6	Thailand political crisis	348	494	1.4
7	Java volcano eruption	47	1,788	38
8	Michael Schumacher's accident	131	1,702	13
9	London Underground Strike	271	6,707	24.7
10	Pussy Riot release	34	5,900	174

TABLE III: Average number of images per OSN and step. The last row presents the ratio of content collected during the second step (M_{ext}) over the content collected in the first (M_0).

OSN	Fb	Tw	G+	Igram	Flickr	Tumblr
$\|M_0\|$	3.6	283.8	31.5	40.7	527.7	9.4
$\|M_{ext}\|$	19.5	813.9	255.8	277.7	2,873	53
$M_{ext}(\%)$	0.45	18.96	5.96	6.47	66.93	1.23
ρ	5.4	2.9	8.1	6.8	5.4	5.6

of media content, which in the case of Tumblr can be explained due to its significantly lower usage (and hence volume of posted content), and in the case of Facebook is attributed to the poor performance of its post search API.

Table III presents the average number of retrieved images per OSN and retrieval step. It appears that Flickr is responsible for more than two thirds (66.9%) of the collected content, and Twitter contributes almost one fifth (≈19%). In terms of increase between the two retrieval steps, several OSNs (Facebook, Flickr, Tumblr) exhibit a five-fold increase. Google+ and Instagram exhibit higher increase rates, 8.1 and 6.8 respectively, while Twitter presents only a threefold increase in the amount of retrieved content.

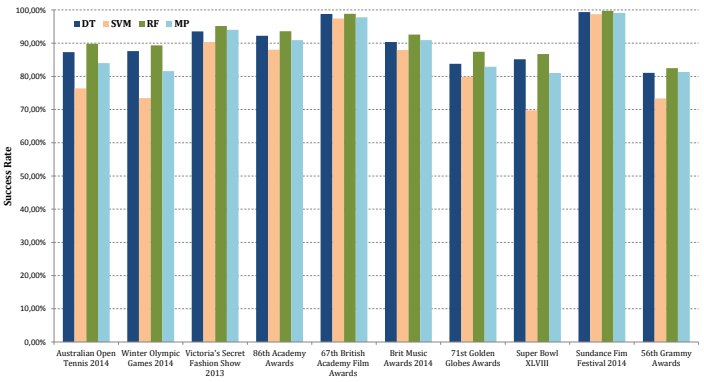
It is noteworthy that searching in Instagram and Tumblr with high-precision queries (M_0) returns almost no content about news stories. Similarly, Flickr high-precision searches for news stories result in considerably less media items compared to events relative to Twitter and Google+. This can be justified by two facts: a) events typically result in much more photos since they last longer and involve numerous scenes and people, while news stories are typically more focused in terms of imagery, b) Twitter and Google+ are much more used for publishing and discussing news stories compared to Flickr.

Quality of formulated queries: The assessment of the queries constructed by the graph-based method described in subsection II-D was carried out through evaluating the relevance and quality of the retrieved content in the second step (M_{ext}). To this end, we conducted a human assessment of the images of M_{ext} with the help of four annotators. Annotation was restricted only to images of M_{ext} since we found that the large majority ($> 90\%$) of images collected during the first step (M_0) were relevant. To this end, we first gave annotators a set of related articles to read about the news story or the event they would annotate, and then a set of specific guidelines on how to decide whether one of the retrieved images is relevant/valuable or not with respect to journalistic interests[2].

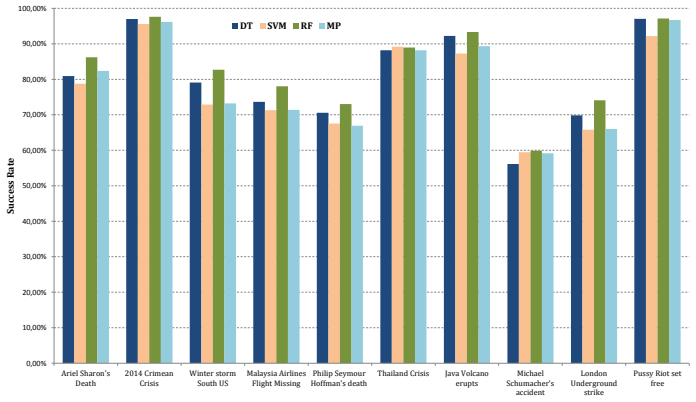
Our analysis indicated that for a small number (three) of events the relevance rate is rather high ($> 50\%$). In the case of news, three news stories exhibit a somewhat lower but decent relevance rate ($\sim 40\%$). Half of the events and news stories are characterized by low-to-medium relevance rates (in the range between 10% and 40%). Finally, for two events and two news stories, the relevance rate is very low ($< 10\%$). Our study revealed that a primary reason for the collection of numerous irrelevant media items is the creation of vague or false keyword-based queries or to the extraction of a vague hashtag. For instance, in the case of the British Academy Film Awards, the most popular hashtag was `british` or in the case of Sundance Film Festival, a vague query `film festival` was formulated. False keyword-based queries are queries that contain irrelevant keywords to the subject and happen mostly because of left-over noisy keywords that the employed graph pruning (cf. subsection II-C) did not manage to eliminate.

Relevance classification: To mitigate the volatility of relevance over different events and news stories, it became necessary to employ a relevance classifier as the one described in subsection II-E. To this end, we evaluated four popular classifiers in combination with the relevance features of Table I: 1) Decision Tree (DT), 2) Random Forest (RF), 3) Support Vector Machine (SVM) and 4) Multilayer Perceptron (MP). All implementations were based on Weka [14], and were evaluated using 10-fold cross-validation. Figures 2a and 2b show the success rate of each classifier in deciding correctly whether a retrieved image is relevant to the topic or not.

Regarding the success rates of the classifiers, Figures 2a and 2b demonstrate that for most of the events and news stories a good distinction between relevant and not relevant images is achieved. As a baseline, one may consider the majority class prediction (i.e. always predict the most likely class), which we computed but cannot present here due to space limitations. In all test cases, the success rate of the proposed relevance



(a) Events



(b) News stories

Fig. 2: Success rate (y axis) of the four tested classification algorithms (DT, SVM, RF, MP) for the events and news stories of Table II.

[2]We became familiar such criteria through our involvement in the news use case of the SocialSensor project: http://socialsensor.eu.

classifier exceeds the baselines (with the exception of the SVM classifier, which performs considerably worse compared to the other three). In many cases, the performance difference is very pronounced, e.g., in the cases of the 86th Academy Awards, where the majority class rate is 58.7% and the success of RF is 93.6%, and the news story about Ariel Sharon's death, where the majority class rate is 55.8%, while the success rate achieved by RF is 86.2%. The comparative study revealed that the RF classifier outperforms the rest in almost all cases. This is closely followed by the DT. The significantly lower rates of the SVM may be attributed to the fact that the input features are not normalized and a few of them quantized to a small set of possible values. As an example of the presented framework output, Figure 2 includes the top 10 images for the 86th Academy Awards.

IV. Conclusions

We examined the problem of searching for multimedia content around events and news stories over multiple OSN platforms. The nature of user-generated content and the large differences with respect to search requirements and behaviour for each platform make it extremely challenging to collect and aggregate high-quality relevant content from multiple OSN sources. We proposed a unified framework that tackles the challenge through a multi-step search process, including a graph-based query building method, and a relevance classification step. The proposed framework was evaluated on a set of 20 large-scale events and news stories of global interest, demonstrating its effectiveness in collecting rich and diverse collections of multimedia content around events and news stories from multiple OSN sources.

In the future, we intend to explore further aspects that were covered in a limited way in this work: a) the degraded performance of the query building method when the volume of media items collected in the first step is small, and means to alleviate it, b) the extraction of relevance features that are statistically grounded, i.e. take into account the differences in the distributions (e.g., of popularity-related variables) arising in different OSN sources, c) the applicability of the framework as an event or news story evolves, and d) the support for collection of video content.

Acknowledgement

This work was supported by the FP7 projects SocialSensor and REVEAL, partially funded by the EC under contract numbers 287975 and 610928, respectively.

References

[1] R. W. Lariscy, E. J. Avery, K. D. Sweetser, and P. Howes, "An examination of the role of online social media in journalists' source mix," *Public Relations Review*, vol. 35, no. 3, pp. 314–316, 2009.

[2] N. Newman, "The rise of social media and its impact on mainstream journalism," *Reuters Institute for the Study of Journalism*, 2009.

[3] L. M. Aiello, G. Petkos, C. J. Martín, D. Corney, S. Papadopoulos, R. Skraba, A. Göker, I. Kompatsiaris, and A. Jaimes, "Sensing trending topics in Twitter," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1268–1282, 2013.

[4] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?," in *Proceedings of the 19th int. conference on World Wide Web*. ACM, 2010, pp. 591–600.

[5] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD int. conference on Management of data*. ACM, 2010, pp. 1155–1158.

[6] M. Nagarajan, K. Gomadam, A. P. Sheth, A. Ranabahu, R. Mutharaju, and A. Jadhav, "Spatio-temporal-thematic analysis of citizen sensor data: Challenges and experiences," in *Web Information Systems Engineering-WISE 2009*, pp. 539–553. Springer, 2009.

[7] D. Metzler and B. Croft, "A markov random field model for term dependencies," in *Proceedings of the 28th annual int. ACM SIGIR conference on Research and development in information retrieval*. ACM, 2005, pp. 472–479.

[8] Y. Lv and C. Zhai, "Positional language models for information retrieval," in *Proceedings of the 32nd int. ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 299–306.

[9] G. Mishne and M. De Rijke, "Boosting web retrieval through query operations," in *Advances in Information Retrieval*, pp. 502–516. Springer, 2005.

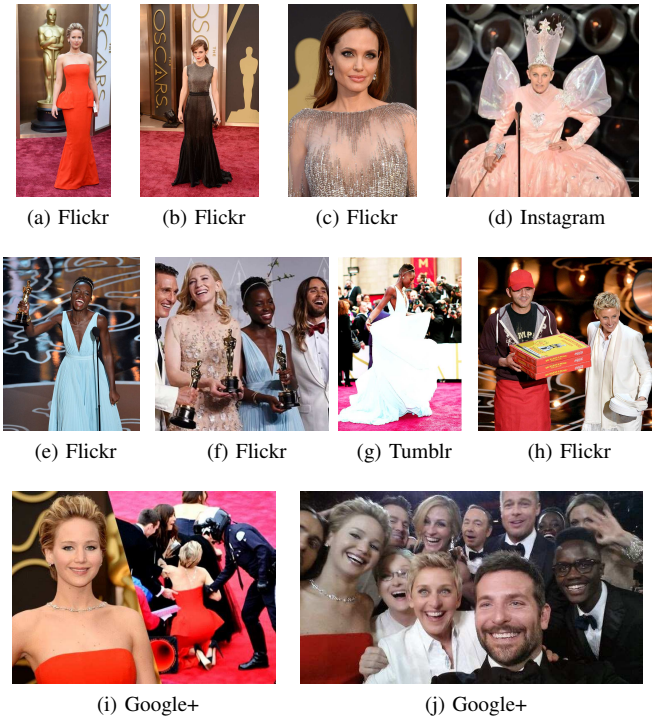
[10] R. Song, M. J. Taylor, J.-R. Wen, H.-W. Hon, and Y. Yu, "Viewing term proximity from a different perspective," in *Advances in Information Retrieval*, pp. 346–357. Springer, 2008.

[11] M. Bendersky, D. Metzler, and B. Croft, "Learning concept importance using a weighted dependence model," in *Proceedings of the third ACM international conference on Web Search and Data Mining*. ACM, 2010, pp. 31–40.

[12] D. Metzler and B. Croft, "Latent concept expansion using markov random fields," in *Proceedings of the 30th annual int. ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 311–318.

[13] E. Spyromitros Xioufis, S. Papadopoulos, Y. Kompatsiaris, G. Tsoumakas, and I. Vlahavas, "A comprehensive study over VLAD and Product Quantization in large-scale image retrieval," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1713–1728, 2014.

[14] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, Burlington, MA, 3rd edition, 2011.



(a) Flickr (b) Flickr (c) Flickr (d) Instagram

(e) Flickr (f) Flickr (g) Tumblr (h) Flickr

(i) Google+ (j) Google+

Fig. 3: Top-10 images for the 86th Academy Awards.