Extracting Resources that Help Tell Events' Stories

Carlo Andrea Conte
Mahaya Inc.
New York, USA
carloante@msn.com

Raphaël Troncy
EURECOM
Biot, France
raphael.troncy@eurecom.fr

Mor Naaman
Cornell Tech and Mahaya, Inc.
New York, USA
mor.naaman@cornell.edu

ABSTRACT

Social media platforms constitute a valuable source of information regarding real-world happenings. In particular, user generated content on mobile-oriented platforms like Twitter allows for real-time narrations thanks to the instantaneous nature of publishing. A common practice for users is to include in the tweets links pointing to articles, media files and other resources. In this paper, we are interested in how the resources shared in a stream of tweets for an event can be analyzed, and how can they help tell the event story. We describe a system that extracts, resolves, and eventually filters the resources shared in tweets content according to two different ranking functions. We are interested in how these two ranking functions perform (with respect to speed and accuracy) for discovering important and relevant resources that will tell the event story. We describe an experiment on a sample set of events where we evaluate those functions. We finally comment on the stories we obtained and we provide statistics that give meaningful insights for improving the system.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

Keywords

Event summarization, Content Analysis, Twitter, URLs

1. INTRODUCTION

For many events and real-world happenings, Twitter, Facebook, Instagram, and other social media platforms provide a continuous stream of user-contributed messages and media. Very often, the messages posted include hyperlinks pointing to content outside the platform where the message was originally posted. The nature of these external resources varies: they may be images, news articles, real-time video streams and many other types of content. Our goal is to identify resources shared via hyperlinks in social media streams that are *highly relevant* and *important* to an ongoing event. Our next goal is to extract these relevant resources, and assemble them in a storyline with the objective of producing a rich narration of the event using a very diverse set of media.

In this paper, we describe a system that aims at extracting a timeline of resources from a stream of tweets[1] about an event. These links will be ranked and filtered in near real-time, in order to identify relevant and valuable information as soon as possible after being shared. In addition, the system extracts descriptive metadata from the referenced pages that can be used to represent the resource in the event timeline in an intelligible way. The collection of items is normalized according to the referenced resources, so that their relevance score will result from the aggregation of the social items referencing them. We will eventually analyse and compare the storylines that are produced in order to identify particular characteristics that could help improve future versions of this system. Our contributions include: a) a general architecture for extracting resources from a stream of tweets; b) the development of two scoring methods to rank the importance of those resources in contributing to the storyline of an event; and c) the representation of meaningful statistics from the resulting story.

The rest of this paper is organized as follow. In Section 2, we present some related work. We then describe our generic system architecture for extracting resources in a stream of tweets (Section 3). In Section 4, we present two ranking methods based on volume and velocity. We show a simple interface we have developed to visualize the storylines that are created based on some filtering parameters (Section 5). We discuss the results of our experiments in Section 6. Finally, we conclude and outline future work in Section 7.

2. RELATED WORK

The system described in this paper is one out of many services that gather social media shared about particular events. These systems generally aim to provide a comprehensive view of an event in order to support the user in following or understanding the event. The primary motivation of our system is to rapidly mine huge volume of social media content so that a user receives relevant information in a timely manner, for example for news reporting or stock market investing.

There has been a lot of work on the exploitation of user

ICMR 2014 SoMuS Workshop, Glasgow, Scotland
.

[1] We observe that tweets can (and often do) contain links pointing to other social networks such as Instagram, YouTube, etc.

generated content and social media for telling events' stories. In this paper, we focus on collecting and mining resources to compose in real-time a storyline about a specific event. Gathering and analyzing contents is the main focus of [1] where the authors provide techniques for automatically identifying posts shared by users on social platforms for a-priori known events, and describe different querying techniques. In [2], the focus is shifted towards identifying the users contributing to the social content available on a particular happening. In [12], the authors select the best tweets for a news article, a related (but somewhat reversed) problem to the one we are presenting here.

Closer to our work, Shamma et al. [9] have looked at summarization and extraction of stories from streams of media, including for example, [5, 10] amongst others. The user-friendly representation of these resources is addressed in [15], where videos provided by the users are assembled in new video streams personalized for each viewer, while in [8], both the retrieval and clustering of media items shared on social networks are assembled in the so-called MediaFinder application. In this paper, we address the problem of building a storyline as soon as the event is progressing.

3. ARCHITECTURE

In this section, we describe the overall architecture for the system that extracts and ranks resources that are linked from Twitter messages about an event. The system we propose needs to run in real-time, in order to build storylines of events as they are unfolding. Our goal is to be able to identify key resources with the smallest delay possible. These requirements impose the adoption of an efficient, flexible concurrency process that would be easily scalable according to the data flow. In the section 6, we report on tests conducted on three events that have happened in the past, with the aim of simulating the same near real-time algorithm that would be applied to happening events.

For the purpose of this work, we assume our input is a stream of Twitter messages that has been identified as relevant to the event being tracked. In our case, these streams are generated by using a hashtag that is associated with each event, but the system described here is agnostic to how the Twitter content is identified. We assume that a separated process retrieves the Twitter content, and keeps updating a database at regular intervals with raw data from Twitter.

The different computational steps performed in the resource extraction process are:

1. Extracting links from a collection of tweets for an event,
2. Resolving these links to their canonical form in order to identify duplicate resources,
3. Ranking links and applying a first basic filter,
4. Collecting useful metadata from the pages referenced by the links which have been selected,
5. Outputting a timeline of resources that can be further filtered using a simple web interface.

We now detail the general architecture and the major building blocks of this system (Figure 1). The complete processing for each link includes two very severe bottlenecks that require network calls that may take a couple of seconds to complete: the resolution of the url, and the scraping of the references pages. In order to overcome this limitation and to build a more efficient system, we split this processing into two systems, that are intended to run in parallel.

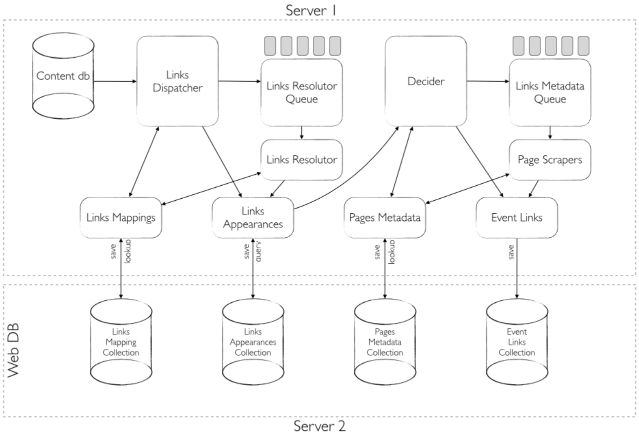


Figure 1: Architecture of the link processing system

The first issue addressed by the system is the fact that, due to URL shorteners and non-canonical URL formats, links to the same resource (e.g. `http://www.example.com/some\_article`) can take different forms: a bit.ly link, a URL that includes a query string, etc. The *links dispatcher* is the first step of this processing chain. It retrieves all the event tweets for a recent temporal window from the *content database*. After loading the raw data, the links dispatcher extracts every link from the tweets' entities [14]), and queries the *links mappings* data to check if the URL has already been resolved to a canonical form. If the link was not yet resolved, the system adds the URL to a *link resolutor queue*. If it was already resolved, the system adds the link, together with the data about the tweet, to the *links appearances* database. The links resolutor queue is implemented as a jobs queue [3]. The advantages of a jobs queue approach is that many workers instances can be run at the same time, and their number and behavior can be adjusted according to the workload. This allows for great flexibility in the way the system can be scaled depending on the amount of data. All the urls in the queue are resolved by the *links resolutor* function. This function will first look up the url in the links mappings. If it is not found, the url will be resolved in order to save a new mapping. For all links which have been successfully resolved, there is an entry in the *links appearances* dataset. The *links appearances* dataset is therefore a collection of all the links which have been resolved. Every item contains information about the url, the tweet that contains this link and the event identifier to which the tweet and the url are attached to.

The second issue is due do the huge amount of resources that are returned which requires to decide and filter what resources will actually be scraped and used to build a storyline. Links appearances are periodically accessed by the *decider*, which is responsible for ranking the resources and filtering them as needed. The computation is always done on a sliding temporal window of a fixed size which is a parameter. The scoring system relies on *links score processors* for its decision process: different score processors can be implemented for testing different score functions. We detail the links score processors (LSPs) used for our experiments in section 4. The LSPs we use implement a very basic filtering in order to enrich the event links with features useful for ranking. Additional filtering possibilities are provided within our front-end interface.

The third issue is caused by the transformation of a set of urls in a more human-readable representation. This requires the scraping of additional information from the pages pointed by the selected links. If a link is selected for publication, the decider queries the *pages metadata* database to check whether the system has the available metadata for that link. If no metadata is available, the link appearance is added to the *Links Metadata Queue*, otherwise it is saved together with its metadata as an *event link*. The *Links Metadata Queue* is processed by the *metadata scraper*. This function extracts the domain from the url and selects a particular scraper class accordingly: it loads the referenced page and extracts pieces of information from it (e.g. a title, a description and a representative image). Different scrapers look for different tags in the DOM structure, as different web sites usually expose different information in different ways. In our tests, we use a generic scraper that collects information stored in the Open Graph[2] and Twitter Cards[3] meta-tags. Only the links for which enough metadata is found are saved as event links, and the pages metadata database is updated accordingly.

The final output of the system is stored in the *event links* database. This dataset holds the final set of resources, together with their score, and other attributes inferred by the link score processor and used to rank the resource (e.g. total volume, highest volume in a time-window duration). In addition, the record contains the metadata extracted from the referenced page.

4. RANKING METHODS

In this section, we describe two methods for ranking the links that appear in the Twitter stream. These methods have to enable a robust detection of relevant and important links with the smallest delay possible from their publication. The reason why we want to filter out irrelevant resources is the quantity of data that is usually shared: it usually largely outgrows the number of items that could be used for building a story.

4.1 Volume based LSP

Organizing a collection of every single appearance of every link gives us the possibility to obtain information about the volume of re-share reached by a link during an event. As an example, Figure 2 shows the number of appearances of the link pointing to president Obama's Vine for Batkid[4].

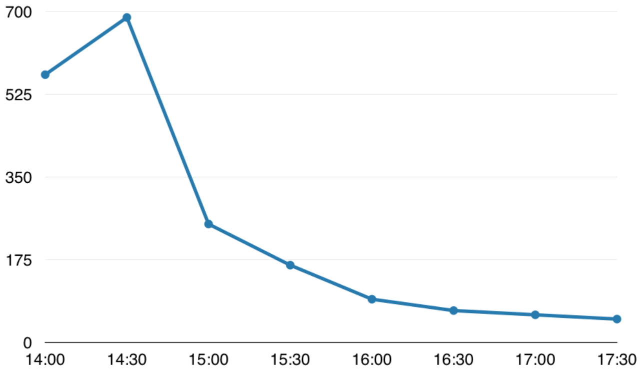


Figure 2: Volume of president Obama's Vine video for Batkid shares

The volume based LSP assigns to every link a score equal to the cumulative function of its volume throughout the event. A very basic filtering step is implemented by a manually chosen volume threshold that is only meant to exclude the noise of links that did not trigger any interest at all in the audience, and can be considered to be background noise (e.g. those links with only one appearance). Such threshold should be set according to the general volume of an event: we heuristically tune these thresholds after extracting from the database aggregated statistics regarding the volume of these links.

A link that has been shared at a nearly constant rate during the analyzed time range will be more likely to appear in our timeline than a link that reached a very high volume at a particular point in time. The display time for links ranked by this LSP is chosen to be the time of the earliest appearance that passed the elementary filtering (e.g. the second appearance of a link). Even if the precision of this parameter will suffer from setting very selective volume filters, we are assuming that high-volume events imply a faster growth of the volume of relevant links, thus introducing only smaller delays. We expect this method to produce more robust rankings. However, this LSP will take longer to recognize important links as the event unfolds, and results will not be reliable while the event is happening as much as when it is over.

4.2 Velocity based LSP

The velocity based LSP computes links' scores as the appearances volume reached by a link within one decider processing time window. The decider's time windows occur every 20 minutes and are 30 minutes long, thus allowing a 10 minutes overlap between each window.

This LSP implements the same filter mechanism described in Section 4.1, with the only difference that the threshold is compared with the current time window's volume. The display time is defined as the first time a link appearance survives the filtering for the first time in a time window. However, the score is always updated to the highest volume the link has reached within a window.

The velocity based method will recognize an important link as soon as it quickly grows in volume, regardless what happened throughout the rest of the event, thus representing a better choice for realtime use. On the other hand, this system can also produce noisier results.

5. FRONT-END INTERFACE

The results of the resource extraction process can be displayed in a simple web interface forming a "storyline": the list of resources are sorted chronologically according to a *display time* field defined by the LSP. Every resource is rep-

[2] `https://developers.facebook.com/docs/opengraph/`

[3] `https://dev.twitter.com/docs/cards`

[4] An initiative of Make-A-Wish Foundation for a child affected by leukemia and that attracted the attention of social media: `http://abcnews.go.com/US/batkids-make-transforming-san-francisco-gotham/story?id=20899254`; President Obama posted a Vine response at `https://vine.co/v/htbdjZAPrAX`

resented by its title, description and an image extracted from the referenced page. We expect this arrangement to automatically provide the reader with a narration of what was happening. This has not been evaluated though. A possible evaluation methodology could consist in generating various timelines according to different thresholds. First, users could be prompted to answer questions related to their *understanding of the event's chronological narration* or the *quality of the storyline* using a Lickert scale. Second, user clicks on timelines could be collected in order to get insights on the number of interesting links included in those summaries. We leave this study as a future work.

A filtering functionality allows a user to select a cutoff score for the links to be visualized. When clicking on a link, that link is marked as false-positive. This marking functionality is used to plot the number of links satisfying a certain volume threshold versus the true-positives satisfying the same requirements. This interface also draws a pie chart, representing the source domains of the links displayed (taking into account the filtering parameter). Figure 3 shows a simple example of a storyline. This example uses the information scraped for each resource to create a visual representation of the content, and an "information feeling" for users to decide whether they would want to click through or not.

6. EXPERIMENTS

In this section, we describe the experiments made to evaluate the efficiency of the storylines built using our method. We aim to compare the results obtained with the two different LSPs ranking functions. We have selected three events that feature very diverse characteristics:

Kanye West at Barclays Center. A concert of the "Resurrects Yeezus" tour that took place in a new New York venue. This event has a relatively low volume of 794 links shared on Twitter during 5 hours. We will refer to it as "#kanye" (Table 1).

Tech Crunch Disrupt. The 2013 conference held in San Francisco. This is a longer event that lasted 3 days (80 hours of content gathered) with a total of 1201 links. We will refer to this event as "#TCDisrupt" (Table 2).

San Francisco's Batkid. An event organized by the Make a Wish Foundation, that transformed San Francisco in Gotham City for one day, letting a child affected by leukemia help Batman to fight the crime. This is a very high volume event, with 8842 links shared, during a timespan of 9 hours. We will refer to it as "#SFBatkid" (Table 3).

For each of those three events, we run the two LSPs scoring functions described in Section 4 using the minimum threshold possible: 1 for #Kanye and #TCDisrupt, and 4 for #SFBatkid in order to obtain a number of links that could be handled by the javascript interface. For #SFBatkid, we extracted the data regarding lower thresholds by directly querying the database.

Media shared on social networks are usually non-permanent and many of the links analyzed by our system are broken. They can either trigger 404 answer (in this case, they are discarded in the process, without affecting the final storyline presentation), or they can point to items which have been removed but for which there is still a page that may contain some metadata. In the latter case, those items will typically appear in the timeline with no description and/or with meaningless titles (e.g. "No Title"). It's important to mention that the older the event becomes, the more likely this type of compromised resource occurs.

#	**Extracted Title**	**URL**
1	Kanye West-Bound 2 (Explicit)	`http://www.youtube.com/watch?v=BBAtAM7vtgc`
2	sashahecht's video on Instagram	`http://instagram.com/p/g65f0pvrJ_/`
3	angelonfire's photo on Instagram	`http://instagram.com/p/g68ZQ1vXXf/`

Table 1: Example of Links from "Kanye West at Barclays Center" by order of appearance

#	**Extracted Title**	**URL**
1	TechCrunch Disrupt Kicks Off with "Titstare" App and Fake Masturbation	`http://valleywag.gawker.com/techcrunch-disrupt-kicks-off-with-titstare-app-and-fa-1274394925`
2	An Apology From TechCrunch|TechCrunch	`http://techcrunch.com/2013/09/08/an-apology-from-techcrunch/`
3	Meet 'Titstare,' the Tech World's Latest 'Joke' from the Minds of Brogrammers-The Wire	`http://www.thewire.com/technology/2013/09/titstare-tech-worlds-latest-brogrammer-joke-techcrunch-disrupt/69171/`
4	And The Winner Of TechCrunch Disrupt SF 2013 Is... Layer!|TechCrunch	`http://techcrunch.com/2013/09/11/and-the-winner-of-techcrunch-disrupt-sf-2013-is-layer/`

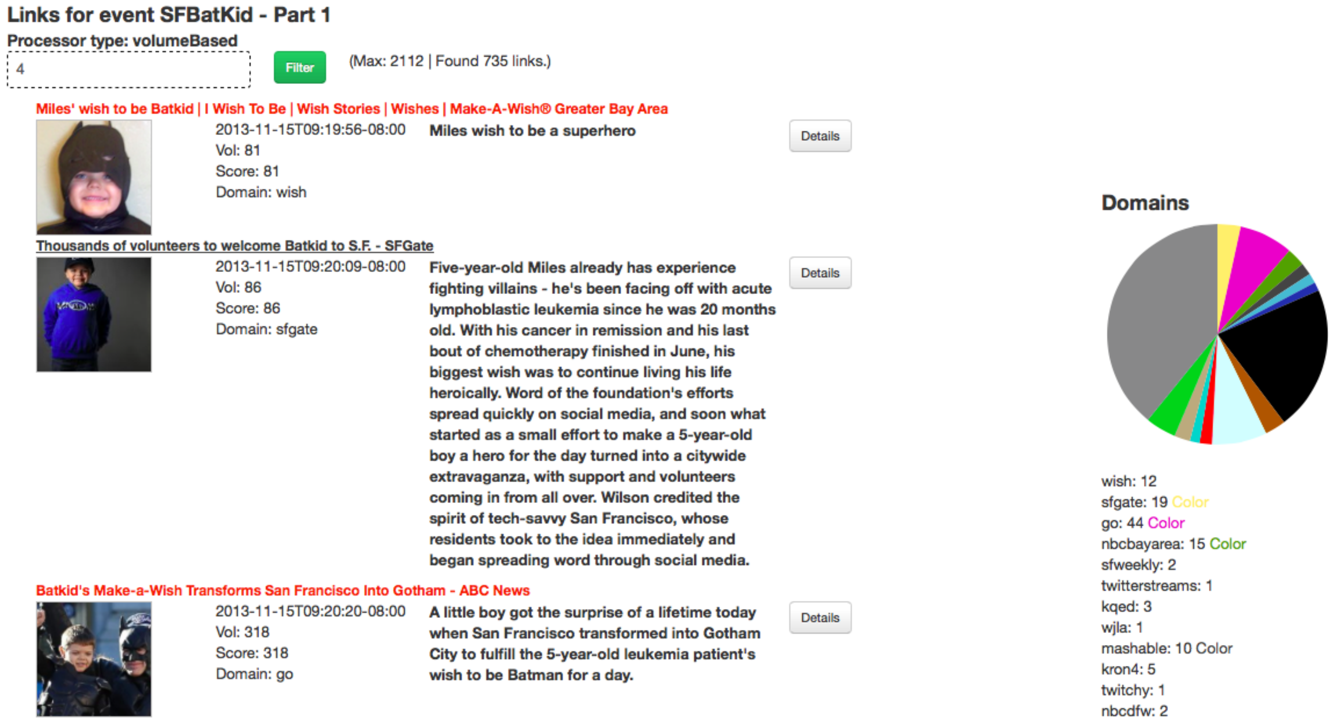
Table 2: Example of Links from "Tech Crunch Disrupt" by order of appearance

When collecting data to compare the number of true-positives against the number of false-positives, we first filter the displayed links according to a volume threshold which is high enough to only select less than 100 links (number that we considered to be optimal for obtaining an enjoyable storyline), and then we proceeded with the false-positives marking process. We always mark compromised resources as false-positives. We also do not consider duplicates to be necessarily marked as false-positives. During our experiments, we also look at the variety of internet domains generating all the links, and the way they varied according to different filter settings. This approach provides useful information for automatically improving the quality of the storylines, for example, by emphasizing on the diversity of the sources.

6.1 Dataset

The data used in our experiments is provided by Seen[5], a service offered by Mahaya Inc. that aims at organizing so-

[5] http://seen.co



Links for event SFBatKid - Part 1

Processor type: volumeBased

4 Filter (Max: 2112 | Found 735 links.)

Miles' wish to be Batkid | I Wish To Be | Wish Stories | Wishes | Make-A-Wish® Greater Bay Area

2013-11-15T09:19:56-08:00
Vol: 81
Score: 81
Domain: wish

Miles wish to be a superhero

Details

Thousands of volunteers to welcome Batkid to S.F. - SFGate

2013-11-15T09:20:09-08:00
Vol: 86
Score: 86
Domain: sfgate

Five-year-old Miles already has experience fighting villains - he's been facing off with acute lymphoblastic leukemia since he was 20 months old. With his cancer in remission and his last bout of chemotherapy finished in June, his biggest wish was to continue living his life heroically. Word of the foundation's efforts spread quickly on social media, and soon what started as a small effort to make a 5-year-old boy a hero for the day turned into a citywide extravaganza, with support and volunteers coming in from all over. Wilson credited the spirit of tech-savvy San Francisco, whose residents took to the idea immediately and began spreading word through social media.

Details

Batkid's Make-a-Wish Transforms San Francisco Into Gotham - ABC News

2013-11-15T09:20:20-08:00
Vol: 318
Score: 318
Domain: go

A little boy got the surprise of a lifetime today when San Francisco transformed into Gotham City to fulfill the 5-year-old leukemia patient's wish to be Batman for a day.

Details

Domains

wish: 12
sfgate: 19 Color
go: 44 Color
nbcbayarea: 15 Color
sfweekly: 2
twitterstreams: 1
kqed: 3
wjla: 1
mashable: 10 Color
kron4: 5
twitchy: 1
nbcdfw: 2

Figure 3: Front-end interface easing the process of extracting and visualizing data.

cial media by building automatic summaries of events [13]. An event can be defined on this system by specifying a set of hashtags and a time range. Once these parameters are known, the contents database (Figure 1) is constantly updated with new raw data gathered from different social platforms at regular intervals in time until the end of the event. No particular filtering on language is performed making Seen a language agnostic service for collecting tweets.

Events' metadata is saved to a collection in a different database (the web databases layer in Figure 1). Initially, a user creating an event in Seen specifies basic metadata for the event being tracked (dates, title, hashtags). Later on, this description is automatically enriched with meaningful information inferred by the service. In our experiment, we only use the metadata specified by the user to retrieve the right subset of raw contents from the contents database. However, we only select events that already exist in the platform, so that we can first acknowledge the general characteristics of each of them in terms of data flow.

6.2 Qualitative Analysis of the Storylines

We provide a qualitative description of the volume based narrations obtained for these three events. Before conducting this analysis, we filtered out using a threshold enabling to reduce the number of items composing a storyline under 100.

6.2.1 *Kanye West at Barclays Center*

We chose a threshold of 2 in order to obtain a storyline of 43 items. Data appears to be noisy, since many links are related to the artist in general instead of this event in particular. Some examples are links to his music video that apparently came out in the same period when the concert

#	Extracted Title	URL
1	SF Morphs Into Gotham City for "Batkid" Battling Leukemia|NBC Bay Area	`http://www.nbcbayarea.com/news/local/SF-Morphs-Into-Gotham-City-for-Batkid-Battling-Leukemia-232054521.html`
2	White House Video's post on Vine	`https://vine.co/v/htbdjZAPrAX`
3	BatKid saves transformed 'Gotham City'-CNN.com Video	`http://www.cnn.com/video/data/2.0/video/us/2013/11/15/dnt-simon-batkid-dream-gotham-city-rescue.cnn.html`

Table 3: Example of Links from "San Francisco's Batkid" by order of appearance

took place (e.g. the first link in Table 1). This problem affects the timeline until around 8PM. At that point, the concert must have effectively started, because between 8PM and 23PM the storyline is only populated by Instagrams depicting various moments of the performance. This strong visual component is a feature that probably characterizes most performance-related events.

6.2.2 *Tech Crunch Disrupt*

All items were filtered with a threshold of 5. The 66 links telling the story of the #TCDisrupt conference are particularly effective in describing what happened at different level of details. They are often news articles coming with very illustrative images, titles and descriptions. The first day of the conference contains most of the links, because it includes a number of general references to the event itself and to the hype for its beginning. The time references seem to be correct: for example, the first item is about the first application that was pitched, and according to some following resources, this project caused a sexist scandal, requiring Tech Crunch to officially apologize (see Table 2).

Further down in the timeline, the links/day ratio shrinks which increases the storyline quality as it mostly includes specific articles about the presentations held on days two and three in chronological order. In particular, the last item closes the storyline by declaring the winner of #TCDisrupt (Row 4 in Table 2).

6.2.3 *San Francisco's Batkid*

This event has a very particular configuration: it contains a huge amount of content (tens of thousands of tweets) shared in a relatively short period of time, mostly as an echo response to mass-medias. The resulting narration, when filtered down to a readable length of 99 items (using a threshold of 32), is very general and redundant. It is mostly composed of articles that describe the event as a whole. Instant media (e.g. Instagrams) has been drowned by the huge number of re-shares achieved by sources such as CNN and NBC. As a result, this timeline is almost exempt from noise but it is much less effective for narrating the event (see Table 3).

6.3 Selection and Ranking quality of Different LSPs

We filtered the storylines resulting from the velocity and volume based processors until we obtained less than 100 items. We marked the false-positive results and we plotted the number of results and the number of true-positives obtained while increasing the threshold. The first important difference we noticed between the two LSPs is the quality of the ranking: while the velocity based LSP tends to concentrate most of the results in the left-most part of the plot (thus in the lower part of the ranking), the volume based one distributes the results better. This strong difference can be seen in Figure 4.

Figure 4 also indicates that our method is efficient in selecting true-positive results when filtering the output of the volume-based LSP. This was not observed with the other two events, where the performance difference between the two LSPs under this point of view was irrelevant. A characteristic of the plot made on velocity-based results is that there is usually a very limited number of highly referenced links that are underlined by their distance from lower-ranked resources. This can be seen in Figure 4 as well as in Figure 5 where the top ranked item is a Vine of president Barack Obama congratulating with the young super-hero, although those few outliers with very high values happened to be, in all our experiments, true positives.

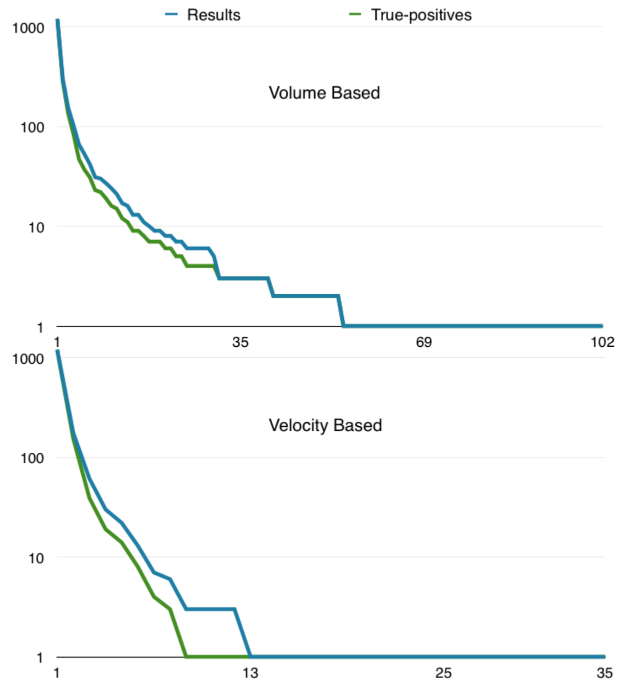


Figure 4: Number of results and number of true-positive results obtained using the volume-based LSP and the velocity-based LSP for the TCDisrupt event while increasing the filtering thresholds

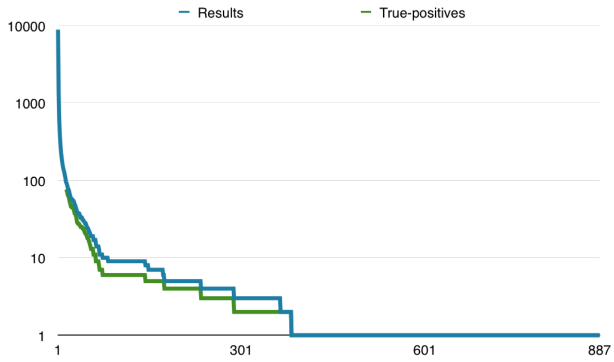


Figure 5: Number of results and number of true-positive results obtained using the velocity-based LSP for the SFBatkid event while increasing the filtering thresholds

The same characteristic is common to all the top ranked elements obtained with the velocity-based method, thus making this selection system a good option for choosing elements to recommend as interesting highlights.

6.4 Domains Composition

The analysis of the source domains underlined how different categories of events can have a very different "fingerprint". Figure 6 clearly shows how a performance-centered event mostly received data from Instagram and Youtube, while a breaking news event and a technological conference are described by a wider variety of newspapers and magazines operating in the respective fields. This information could be used to automatically detect events' categories, or to implement a smarter ranking function that assigns different importance to links coming from different sources, when the category of the event is known. We also noticed how some of the biggest generators of social content (i.e. Instagram, Facebook) tend to disappear from the pie chart when rising the volume threshold above one. This underlines the importance of an additional dimension, the volume, in defining a "category fingerprint" in this particular space of the source domains.

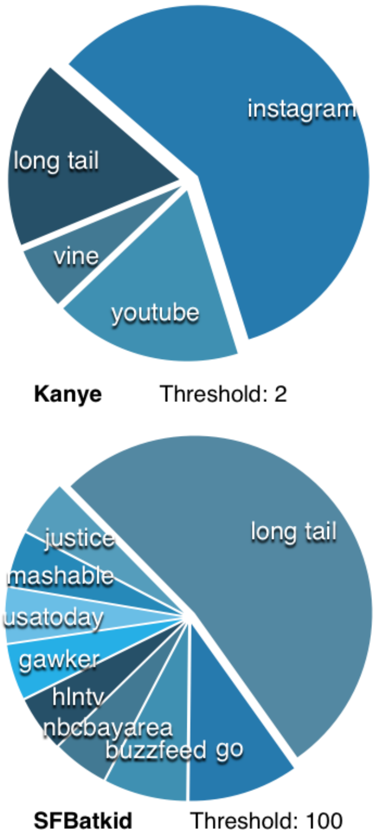


Figure 6: Source domains composition for the SFBatkid and the Kanye events. Volume thresholds have been chosen as the highest that still allowed enough results to produce a meaningful analysis

This analysis can also help to automatically identify official sources for a given event. In fact, if the category fingerprint of an event is given, official sources can sometimes be identified as outliers: this can be clearly seen in Figure 7 where the "techcrunch" domain has a remarkably outsized cardinality comparing with the other ones.

Our original goal was to explore the data produced by the system we have implemented. Therefore, while the results we are reporting are constrained by the settings we have chosen, they well serve the purpose of unveiling interesting patterns that should be further investigated by experimenting on larger sets of events.

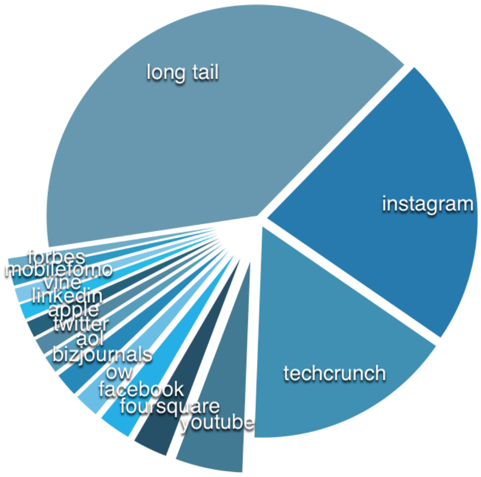


Figure 7: Source domains of the TCDisrupt event computed on results obtained without any filtering

7. CONCLUSION AND FUTURE WORK

In this paper, we presented a system for transforming links shared on social media into narratives of what happened at specific events. These narratives are composed by the set of pages referenced in these links, filtered according to a score. We evaluated the storylines obtained and we extracted some information that can help designing new methods to improve their quality. The collection of resources produced by this method imposes the analysis of social activity in a new space, where the single tweets shared are aggregated into a feature of the resources they reference. Not only this approach can produce real-time meaningful narratives (as these resources become very descriptive thanks to the page scraping process), it can also help extracting useful insights, for example the composition of the source domains.

We noticed how different characteristics of an event can affect the efficiency of this system: while we obtained a good narration of a technology conference, the results obtained for a breaking news event were more disappointing. Further research should be conducted on this topic, in order to define some better score functions tailored on the characteristics of each category of event (e.g. considering alternatives to volume based score functions for a breaking news event). We also defined a useful feature based on the evolution of the composition of the source domains with increasing volume thresholds. This can help identifying the category of an event and it could also provide the information necessary to automatically identify official sources, when these are particularly active on social channels.

Acknowledgments

The research leading to this paper was partially supported by the European Union's 7th Framework Programme via the projects LinkedTV (GA 287911) and MediaMixer (GA 318101).

8. REFERENCES

[1] H. Becker, D.Iter, M. Naaman, and L. Gravano. Identifying Content for Planned Events Across Social Media Sites. In 5^{th} *International ACM Conference on*

Web Search and Data Mining, Seattle, Whashington, USA, 2012.

[2] M. D. Choudhury, N. Diakopoulos, and M. Naaman. Unfolding the Event Landscape on Twitter: Classification and Exploration of User Categories. In *15th ACM Conference on Computer Supported Cooperative Work*, Seattle, Whashington, USA, 2012.

[3] V. Driessen. Redis Queues Python Library. `http://python-rq.org`, 2013.

[4] A. Joly, J. Champ, P. Letessier, N. Hervé, O. Buisson, and M. Viaud. Visual-Based Transmedia Events Detection. In *20th ACM international conference on Multimedia (MM'12)*, Nara, Japan, 2012.

[5] Y.-R. Lin, H. Sundaram, M. D. Choudhury, and A. Kelliher. Temporal Patterns in Social Media Streams: Theme Discovery and Evolution Using Joint Analysis of Content and Context. In *IEEE International Conference on Multimedia and Expo (ICME'09)*, pages 1456–1459, Piscataway, NJ, USA, 2009.

[6] V. Milicic, J. L. Redondo García, G. Rizzo, and R. Troncy. Tracking and Analyzing The 2013 Italian Election. In *10th Extended Semantic Web Conference (ESWC'13), Demo Session*, Montpellier, France, 2013.

[7] V. Milicic, G. Rizzo, J. L. Redondo García, R. Troncy, and T. Steiner. Live Topic Generation from Event Streams. In *22nd World Wide Web Conference (WWW'13), Demo Session*, Rio de Janeiro, Brazil, 2013.

[8] G. Rizzo, T. Steiner, R. Troncy, R. Verborgh, J. L. Redondo García, and R. V. de Walle. What Fresh Media Are You Looking For? Retrieving Media Items from Multiple Social Networks. In *International Workshop on Socially-aware multimedia (SAM'12)*, Nara, Japan, 2012.

[9] D. A. Shamma, L. Kennedy, and E. F. Churchill. Conversational Shadows: Describing Live Media Events Using Short Messages. In *4nd International Conference on Weblogs and Social Media (ICWSM'10)*, Washington, USA, 2010.

[10] D. A. Shamma, L. Kennedy, and E. F. Churchill. Peaks and Persistence: Modeling the Shape of Microblog Conversations. In *International Conference on Computer Supported Cooperative Work (CSCW'11)*, pages 355–358, New York, NY, USA, 2011.

[11] T. Steiner. A Meteoroid on Steroids: Ranking Media Items Stemming from Multiple Social Networks. In *22nd World Wide Web Conference (WWW'13), Demo Session*, Rio de Janeiro, Brazil, 2013.

[12] T. Łtajner, B. Thomee, A.-M. Popescu, M. Pennacchiotti, and A. Jaimes. Automatic Selection of Social Media Responses to News. In *19th International ACM Conference on Knowledge Discovery and Data mining (KDD'13)*, Chicago, Illinois, USA, 2013.

[13] R. Tate. The Next Big Thing You Missed: Recreate Live Events with Twitter and Instagram. `http://www.wired.com/business/2013/11/seen-is-real-life-instant-replay/`, 2013.

[14] Twitter. Twitter's REST API Documentation - Tweets. `https://dev.twitter.com/docs/platform-objects/tweets`, 2014.

[15] V. Zsombori, M. Frantzis, R. L. Guimaraes, M. F. Ursu, P. Cesar, I. Kegel, R. Craigie, and D. C. A. Bulterman. Automatic Generation of Video Narratives from Shared UGC. In *22nd ACM Conference on Hypertext and Hypermedia*, Eindhoven, The Netherlands, 2011.