# Mining Events Connections on the Social Web: Real-Time Instance Matching and Data Analysis in EventMedia

Houda Khrouf, Vuk Milicic, Raphaël Troncy *

*Multimedia Communications Department, EURECOM, 06410 Sophia Antipolis, France*

**Abstract**

Event and media services have recently witnessed a rapid growth driving the way people explore information of interest. A significant amount of social calendars, media memes and background knowledge are daily created on various platforms, conveying event clues or past users experience. Mining, in real-time, the connection of these distributed data fragments provides a key advantage not only to deliver enriched views, but also to gain insight into interesting sociological aspects. To this aim, we harness the power of Semantic Web technologies as means to easily steer the data integration and analysis. Our overall goal is to build a web-based environment that allows users to discover meaningful, surprising or entertaining connections between events, media and people.

In this paper, we present EventMedia, a platform that provides descriptions of events associated with media, and interlinked with the Linked Data cloud. It draws on a live data update and a real-time interlinking to face the natural dynamics of events. A user-friendly interface has been designed to meet the user needs: relive experiences based on media, and support decision making for attending upcoming events.

*Key words:* EventMedia, LODE ontology, real-time, instance matching, event-based social network

## 1. Introduction

The social event landscape is increasingly crowded with new web sites including media platforms, social networks and event directories. People have been recently attracted by these services to organize their personal data according to occurring events, to share captured media and to express their thoughts. Web sites such as Eventful, Upcoming, Last.fm or Flickr host an ever increasing amount of event-centric knowledge maintained by rich social interactions. The problem is that this knowledge represents a large space of disconnected data fragments providing limited event coverage [3]. For instance, while Last.fm sustains a broad coverage on people attending events, it is only limited to musical concerts and often provides few details about tickets and media. We believe that all these social channels could mutually complement each other in order to produce a rich content and promote event awareness.

In this paper, events considered are a natural way for referring to any observable occurrence grouping persons, places, times and activities. They represent observable experiences that are often documented by people through different media [12]. Our belief is that the spatial-temporal dimension, the human participation, the illustrative media and the background knowledge are meaningful components to enhance content views. This work takes initial steps towards building EventMedia, a platform that ag-

* Corresponding author. Tel: +33 (0)4 - 9300 8242
*Email addresses:* houda.khrouf@eurecom.fr (Houda Khrouf), vuk.milicic@eurecom.fr (Vuk Milicic), raphael.troncy@eurecom.fr (Raphaël Troncy).

gregates and interlinks in real-time heterogeneous data sources leveraging on the benefits of Semantic Web technologies. We use some heuristics to mine the intrinsic connections of event-centric data from event directories, media platforms and Linked Data. In particular, the transiency of events imposes a challenge to maintain a real-time data crawling and reconciliation which will ensure a dynamic content enhancement. In addition, we conduct an analysis to highlight the role of data reconciliation to detect some social aspects about users' participation.

The remainder of this paper is organized as follows: Section 2 describes the results of a user study designed to assess the quality of existing technologies. Section 3 provides an overview of our system architecture. Section 4 outlines the approach applied to reconcile data, and we describe experiments in Section 5. We highlight the benefits of data reconciliation through an analysis of social factors in Section 6. Finally, we describe the user interface in Section 7 and we conclude in Section 8.

## 2. User-centric Study

The motivation behind this work has been proved through an exploratory user centric study conducted to assess the perceived qualities of available event and media directories [14]. This study consisted of a user survey completed by 28 participants, and two focus-group sessions (10 and 25 participants). It was carried out to understand end-users' event-related experiences, and to collect insights about existing web-based technologies that support related activities. As a result, lack of coverage of event directories and frustration of being locked in a particular site or social network are the recurrent issues. Participants recognized that there was a need to access several social channels to gather information. One participant reported *I don't like always having to go from one site to another to find out things about the event.* Overall, users advocate the need for a single source to explore events, not by creating another information source, but by centralizing all available information leading to broader coverage. In addition, they highlight the role of images and videos to provide powerful means for identifying several event characteristics, to convey the experience and to support decision making. Nevertheless, a common concern of information overload suggests that the environment should avoid cluttered information and provide browsing options to meet the user constraints. Motivated by this perspective, we decided to build a platform based on Semantic Web technologies to merge information spread in many silos and enhance the event coverage.

## 3. System Architecture

EventMedia is a hub in the Linked Data cloud since September 2010. It is obtained from three public event directories (Last.fm, Eventful, Upcoming) and from one large media directory (Flickr). It encapsulates media and events descriptions, enriched with background knowledge from external datasets such as DBpedia, MusicBrainz, BBC and Foursquare. EventMedia is based on LODE ontology [12] and consists of more than 30 millions RDF triples. All URIs are dereferencable and served as either static RDF files serialized in N3 or as JSON by a RESTful API. The back-end of the system consists of a live data crawler and an interlinking framework, a RESTful API powered by the ELDA implementation of the Linked Data API[1] and a Virtuoso SPARQL endpoint[2]. A complete system architecture is depicted in Figure 1.

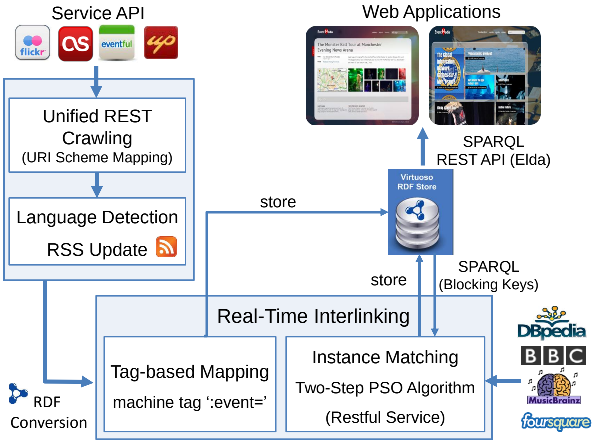


Fig. 1. EventMedia System Architecture

### 3.1. *Data Crawling*

Data crawling of various web API is a complex and time-consuming task due to the heterogeneous services specifications. To mitigate this issue, a key idea is to leverage the commonalities of these APIs, so that they could be exported into one unified

[1] http://code.google.com/p/linked-data-api
[2] http://eventmedia.eurecom.fr/sparql

RESTful service. Such service should be able to deal with many tasks such as policy management, requests chaining, data integration and merging response schemas. Thus, we built a framework that queries multiple web APIs, namely Eventful, Upcoming, Last.fm and Flickr, and converts events and media descriptions into a unified data model [7]. The framework is composed of a REST-based module that defines new methods and maps them with associated methods in targeted web APIs. For instance, the method for collecting events takes as input a set of parameters such as source (e.g. Eventful, Upcoming, etc.), category, location and dates. Thus, a user is able to request in parallel multiple sources using a single query. The output is processed by a second module performing several tasks starting from JSON de-serialization, language detection to RDF conversion and loading into a triple store. Finally, a web dashboard has been designed to easily handle queries for data crawling. The system also offers an interface to reconcile data and to have detailed statistics about the dataset. It is accessible online at `http://eventmedia.eurecom.fr/dashboard`.

### 3.2. *Data Modeling*

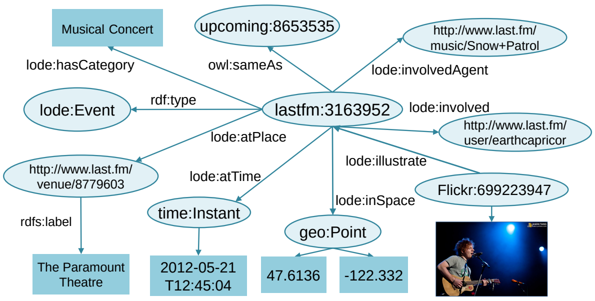


Fig. 2. *Snow Patrol Concert* described with LODE ontology

Once collected, data is converted into RDF triples providing description of events using the LODE ontology and a large SKOS taxonomy of event categories. LODE is a minimal model that encapsulates the most useful properties for describing events. The goal of this ontology is to enable interoperable modeling of the "factual" aspects of events, where these can be characterized in terms of the four Ws: What happened, Where did it happen, When did it happen, and Who was involved. The dataset contains a highly diverse set of categories, ranging from large festivals and conferences or exhibitions to small concerts and social gatherings. It also provides the description of media using the W3C Ontology for Media Resources and uses properties from SIOC, FOAF, Dublin Core and vCard. Figure 2 depicts the metadata attached to the event identified by `3163952` on Last.fm according to the LODE ontology. More precisely, it indicates that an event of type `Concert` has been given on the `21th of May 2012 at 12:45 PM` in the `The Paramount Theatre` featuring the `Snow Patrol` rock band, and one attendee is the Last.fm user `earthcapricor`. This event is matched with a similar one announced on Upcoming.

## 4. Real-Time Interlinking

At the core of our system is the real-time reconciliation framework that aligns every incoming stream of overlapping but highly heterogeneous data sources. This will sustain a continuous content enhancement, a crucial task to cope with the dynamics of social services. The major gain is to bring valuable information and expand the reach of an event at different stages. In fact, viewing an event page from one event-based service underlines an incomplete content that needs to be further enriched. For instance, we have always detected a real lack of involved agents and their descriptions in the Upcoming directory, whereas in Last.fm, people are more responsive to attend events, but only limited to musical concerts without complete description. We believe that reconciling event-centric data will mutually leverage the benefits of each service and achieve a better event overview. Hence, we first explore the connections between events and media using tag-based mapping. Then, we propose a RESTful service to mine in real-time meaningful connections between events, agents and locations in Linked Data based on instance matching techniques.

### 4.1. *Tag-based Matching*

We explore the overlap in metadata between Flickr as a hosting web site for photos, and the event-based services. We interlink every incoming stream of photos with associated events using an explicit relationship materialized by a machine tag such as `lastfm:event=event_id`. Hence, we have been able to convert the description of more than 2 million photos which are indexed by nearly 160.000 events. Other machine tags have also been exploited to establish `owl:sameAs` links with Foursquare and Musicbrainz.

### 4.2. *Correlation Driven Instance Matching*

Instance matching is a well-known task of central importance in Linked Data. It aims to discover identity relationships among structured data to construct *owl:sameAs* links between same real-world objects. We employ instance matching to mine connections between heterogeneous sources for different instance types, namely: event, agent and location. There is, therefore, a strong need for domain-independent matching to overcome the diversity of vocabularies used. Moreover, the likelihood to encounter different object values or typographical errors is higher in event repositories rather than, for example, in encyclopedic web sites. We particularly noticed that some of semantically dissimilar properties can have a latent relationship valuable for instance matching. For example, the `dc:title` of one Last.fm event is *"Cale Parks at Pehrspace"*, whereas the `dc:title` of the same Upcoming event is *"Cale Parks, The Flying Tourbillon Orchestra, One Trick Pony, Meredith Meyer"* which lists all involved agents, rather expressed via `lode:involvedAgent` in Last.fm. Such particular heterogeneity have been rarely addressed in existing matching tools that mostly utilize manual definitions of property pairs or attempt to link properties having similar semantics such as `dc:title` and `rdfs:label`.

Among the instance matching tools, Silk [4] draws on a declarative configuration language with which the user must manually define the properties of each instance to be compared. However, when setting these properties, we mainly follow our intuition and we may skip useful latent similarities between other predicates. Zhisi.Links [11] is two-step matching that firstly remove highly different instances using the label of objects, and secondly utilize a semantic metric. One drawback of a typical system is to rely on the labels of objects as candidate selection key, which is only convenient when there is an exact match between two labels. To solve this, Song et al. [13] are based on unsupervised leaning to propose a blocking scheme based on the discrimination and the coverage of predicates. Although this approach is interesting, it is mostly biased to string literals and no consideration for other data types was made. Another unsupervised approach has been proposed in KnoFuss [10] which exploits genetic programming to discover the required similarity parameters with a focus on precision maximization.

Different from the previous studies, we consider the various data types and define a new supervised blocking scheme based on the correlation and the coverage of predicates. Like Zhisi.Links [11] and Song et al. [13], our two-step approach uses a blocking key for candidate selection in the first step, but it uses a training method to discover the best weights of similarity function in the second step. To compute the similarity between object values, we exploit various metrics according to data types such as string, date-time and numeric.

***String.*** For string data type, we lowercase the literals and filter the stop-words. To compute similarity, we use Cosine distance enhanced by Porter stemming for long strings (e.g. description) and Token-Wise [6] distance for short strings. Token-Wise is a hybrid metric combining character-based (e.g. Levenshtein [8]) and token-based functions, useful to overcome typographical errors and ignore tokens ordering. Considering equal weights for all tokens (i.e. no penalized tokens) and retaining the highest character-based score for each token, Token-Wise is measured as following:

$$sim(S,T) = \frac{\sum_{s \in S, t \in T} \ Levenshtein(s,t)}{max(|S|,|T|)} \qquad (1)$$

where S and T are the token sets of the compared strings.

***DateTime.*** One main concern to discover same real-world events is to consider not only the distance between two date-time values, but also the inclusion of a date-time value in a period of time or the temporal overlap between two periods of time. Thus, we define a novel temporal inclusion metric [6] that computes the temporal difference, the inclusion or the overlap depending on the availability of start/end dates of two events. More precisely, we detect whether two events have close dates or whether they share a common temporal interval. The metric returns either 0 or 1 and can tolerate a certain number of hours predefined by the user. For instance, the metric returns 1 when computing the similarity between an event given from `30th at 08:00 PM to 31th at 07:00 PM of May 2012`, and another event given on `31th May 2012 at 9:00 PM` with 2 tolerated hours.

***Numeric.*** We simply compute the reciprocal of the absolute value of the difference between two numeric object values.

To gain insights into which property pairs worthy to be compared, we lean on the correlation and the coverage factors measured from labeled data. These factors will also discern the candidate selection key used to maximize the coverage of true matches in the first step of our approach. We take as input two matched instance sets $I_s$ (source) and $I_t$ (target). For each set $I_i$ ($i \in \{s,t\}$), we retrieve the set of literal values $L_i$ associated with each property $p_i$ at a distance *n-path* from individuals in $I_i$. If a property is used more than one time, we group the associated multiple values into one value. The correlation reflects the mutual information shared between two properties from the source and the target sets. Each data type of $L_i$ is associated with similarity function $sim_d$ explained above, except for string literals we only use Cosine distance. We formalize the correlation and the coverage of each property pair as follows:

$$Corr(p_s, p_t) = \frac{\sum_{l_s \in L_s, l_t \in L_t} sim_d(l_s, l_t)}{min(|L_s|, |L_t|)} \tag{2}$$

$$Cov(p_s, p_t) = \frac{min(|L_s|, |L_t|}{|I_s|} \tag{3}$$

Useless predicates having very low correlation and coverage are filtered out. We consider that the selection candidate key is formed from the predicates which exhibit high correlation and maximum coverage. Then, the remaining properties are used to compute the overall similarity score. We explain how to compute similarity in Section 5 which details our experiments. As a training method to find out the weights and the threshold of similarity function, we employ the Particle Swarm Optimization (PSO) [5], a population based stochastic technique inspired by the social behavior of bird flocking or fish schooling. The algorithm initializes a population of random solutions called particles, and searches for optima of a fitness function by updating generations. In each generation, each particle accelerates in the direction of its own personal best solution found so far, as well as in the direction of the global best position discovered so far by any of the particle in the swarm. In our approach, a particle is represented by a vector of weights and thresholds, and the fitness function aims at maximizing the F-score.

***Real-Time.*** A fundamental concern in EventMedia is the real-time data reconciliation to efficiently cope with the growing amount of events daily created on the social web. To achieve this, we build a RESTful service that allows to run instance matching of freshly stored data. More precisely, the service retrieves data from the triple store by means of two kinds of SPARQL queries. The first query fetches the set of instances of the source dataset filtering the data collected using `rdf:type` and the start/end storage dates (expressed via `dc:issued`). The second query retrieves, for each instance, a set of candidate solutions from the target source using the `rdf:type` predicate and some filters determined by the highest correlated and frequent predicates. For instance, to retrieve the candidate solutions for one person's name, we use each unigram of its label as filters (more details in Section 5).

## 5. Experiments and Results

In this section, we describe a set of experiments to align the resources of type: Event, Agent and Location. We demonstrate the effectiveness of our method by means of two ground truths. The PSO parameters used for the experiments are population size=25, iterations=40, acceleration coefficients $c_1$=1.494 and $c_2$=1.494, and inertia weight $w$=0.729 (recommended setting of $c_1$,$c_2$,$w$ in [2]). Statistics about the resulting linksets are accessible on the dashboard.

### 5.1. *Events Matching*

We define events similarity as a mutual agreement in terms of their factual properties, namely: title (what), time (when), location (where) and involved agents (who). Nevertheless, the correlation and the coverage factors of associated predicate pairs depend on the source and the target datasets. Herein, we focus on matching events derived from Last.fm and Upcoming web sites, and we evaluate it using a manually constructed ground truth containing 300 events (happened in 2009). We retrieve literal values at a distance 3-path due to the presence of blank nodes. In Table 1, we report the high to fair correlated properties (correlation $\geqslant 0.3$) computed with 100 events, widely enough to implicitly recognize which predicates are important.

From Table 1, it can be noticed that both *time* and *place* properties exhibit a total coverage and a high correlation, thus their combination forms the candidate selection key. Note also that a significant correlation exists between $agents_s$ and $title_t$ corresponding to semantically dissimilar properties, but conveying a connotative relationship. To select candidates for each instance in $I_s$, we retrieve the en-

| $P_{source}$ | $P_{target}$ | *Correlation* | *Coverage* |
|---|---|---|---|
| $time_s$ | $time_t$ | **1** | **1** |
| $place_s$ | $place_t$ | **0.80** | **1** |
| $title_s$ | $title_t$ | 0.59 | **1** |
| $agents_s$ | $title_t$ | 0.53 | **1** |
| $(lat_s, long_s)$ | $(lat_t, long_t)$ | (0.43, 0.97) | 0.92 |

Table 1
Correlation and Coverage rates between properties of 100 events from Last.fm and Upcoming

tities from $I_t$ which are associated with an average score of key predicates $(time, place)$ greater than a threshold $\alpha$. Yet, considering this key in the real-time scenario is not trivial, since SPARQL does not support a complex arithmetic computation (e.g Levenshtein). As a solution, to retrieve interesting candidates with SPARQL, we decided to use the *time* predicate in conjunction with a full-text search applied on each token from *place* and *title* values (after removing stop-words).

To evaluate our approach, we conducted various tests comparing the pure weighted linear combination of similarity scores (LC), and the methods using the candidate selection such as the two-step linear combination and the two-step boolean reasoning (OR). At the time of writing, we were not able to find an independent-domain matching tool that overcomes structural heterogeneity with a real consideration of all data types (without bias to string literals). For this experiment, we choose to compare our approach with KnoFuss [10], an independent-domain tool based on genetic algorithm (GA) to discover the components of the best similarity decision including the property pairs, the metrics, the weights and the threshold. We integrated our metrics, namely the token-wise and the temporal inclusion in KnoFuss, and we report the results in Table 2.

| | *Precision* | *Recall* | *F-score* |
|---|---|---|---|
| LC KnoFuss (GA) | 0.94 | 0.74 | 0.83 |
| LC (PSO) | 0.88 | 0.96 | 0.92 |
| Two-step LC (PSO) | 0.91 | 0.95 | 0.93 |
| Two-step OR (PSO) | **0.96** | **0.97** | **0.96** |

Table 2
Results of different approaches to align events between Last.fm and Upcoming (50% training data)

We can observe that KnoFuss yields high precision but the lowest recall. This is owed to its strategy that maximizes a pseudo F-score with bias to precision optimization, given that the cost of an erroneous mapping is higher than the cost of a missed correct mapping. It is also shown that the two-step methods produce better results than the pure LC methods, due to the effectiveness of candidate selection key to remove noisy information. In particular, the two-step OR method uses the key $(time + place)$ to filter candidates. Then, we assume that it is sufficient whether one score obtained from the remaining highly/fairly correlated properties is larger than a trained threshold. This method outperforms the other LC-based methods, since it succeeded to overcome the lack of coverage of latitude/longitude predicates. Indeed, in the he LC-oriented methods, the weight assigned to the geographical distance is very low due to the limited coverage of these predicates, whereas a high weight was assigned by the OR-oriented method. Finally, Table 3 details the results obtained by the Two-step OR method for different training splits. It is clear that this method achieves a good performance even for small training set.

| | *Precision* | *Recall* | *F-score* |
|---|---|---|---|
| 30% | 0.95 | 0.96 | 0.95 |
| 50% | 0.96 | 0.97 | 0.96 |
| 80% | **0.99** | **0.98** | **0.99** |

Table 3
Resuls of Two-step OR algorithm for event alignment with different splits of training data

For events matching, we also investigate the connection between EventMedia and DBpedia. We note that both datasets encapsulate the description of events, but different in terms of data model and data granularity. Indeed, EventMedia provides a fine-grained information detailing a spatio-temporal dimension along with other properties. Conversely, DBpedia keeps a general level of description of very famous events without a granular precision about the event time, except for few of them. Considering this fact, we decided to create `rdfs:seeAlso` links between events from these datasets, producing 1-*to*-$N$ mapping instead of 1-*to*-1 mapping. To achieve this, we use SPARQL queries and label-based pattern-matching by setting a high threshold.

### 5.2. *Agents Matching*

Mining agents connections plays an important role to bring valuable context such as artists' discography, fine detailed biography and illustrative photos. We reconcile agents derived from event-based

| $P_{source}$ | $P_{target}$ | *Correlation* | *Coverage* |
|---|---|---|---|
| $label_s$ | $label_t$ | 0.69 | 1 |
| $subject_s$ | $genre_t$ | 0.52 | 0.90 |
| $description_s$ | $comment_t$ | 0.35 | 0.98 |

Table 4
Correlation and Coverage rates between agent properties from Last.fm and DBpedia

services, and with open datasets such as DBpedia, Musicbrainz[3] and BBC[4]. Since the agents' names exhibit the highest correlation and the total coverage, we consider each name token as a blocking key to fetch similar candidates using SPARQL full-text search. In this context, the key challenge widely investigated in the literature is to resolve the naming conflicts. Thus, we invoke additional information using the fairly correlated properties such as `dc:subject` and `dc:description`. Table 4 shows the correlation and the coverage rates measured on 100 agent pairs between Last.fm and DBpedia. In this experiment, we point out many correlated properties having the same meaning since DBpedia do not use one vocabulary. For instance, a person name is represented by three properties `rdfs:label`, `foaf:name` and `dbprop:name`. Hence, we manually select the most correlated properties having different meaning. Using a ground truth of 2000 agent pairs from Last.fm and DBpedia, the two-step OR method achieves the best performance with F-score=0.98 (precision=0.99, recall=0.98).

### 5.3. *Venues Matching*

Venues matching was particularly straightforward owing to the consistent and complete description represented by a set of fields such as address, geo-coordinates, city, postal-code and country. We reconcile venues derived from event-based services, and with external datasets such as Foursquare and DBpedia. We did not build a ground truth for this task, but we found that the similar instances checked on the fly are correctly matched. Moreover, a significant number of venues have been reconciled especially with the Foursquare repository.

### 5.4. *Real-time Processing*

To ensure the real-time processing, we created a scheduler that executes two successive tasks every 10-minutes: (i) the first task enables to fetch new photos in Flickr feeds (size of 20 items) and trigger accordingly the scraping requests to retrieve photos and events description; (ii) the second task aligns the freshly stored data with various sources by sending HTTP requests to the reconciliation framework. To evaluate the real-time scenario, we take a sample of data collected during 3 days and we compute two measures, namely: the storage interval which is the difference between the time data is uploaded in Flickr and the time data is stored in the triple store, and the reconciliation interval which is the difference between the time data is stored in the triple store and the time data is reconciled.

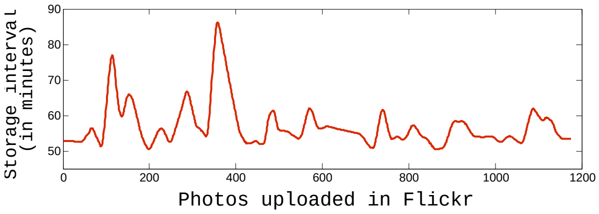


Fig. 3. Evaluation of the storage interval

The response time of one scraping task related to one RSS feed ranges from few seconds to 3 minutes. This duration is mainly affected by the number of the event-related entities such as artists and attendees, which require each an API request. In Figure 3, we observe that the storage interval varies from 50 to 90 minutes attesting that our system contains the freshly uploaded photos in Flickr. We note that this variation is correlated with the delay between uploading photos and updating the Flickr RSS.

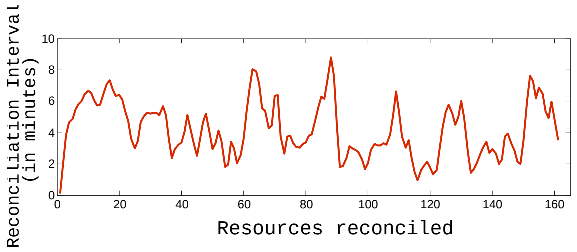


Fig. 4. Evaluation of the reconciliation interval

The response time of one reconciliation task ranges from few seconds to 6 minutes depending on the number of entities to be reconciled. Figure 4 highlights the short interval between the storage

[3] `http://musicbrainz.org`
[4] `http://www.bbc.co.uk`

and the matching times, which approves the efficiency of our real-time reconciliation strategy.

## 6. Social Data Analysis

In this section, our main goal is to address the following research question: apart from enriched views, what the interlinking is valuable for? Looking for an answer, we decided to carry out an analysis of non-exhaustive social factors covering various aspects such as the tendency of sharing photos and attending events, and the user interests. In this context, we leverage the Linked Data technologies as particularly powerful means to access data.

### 6.1. *How Many Photos Shared?*

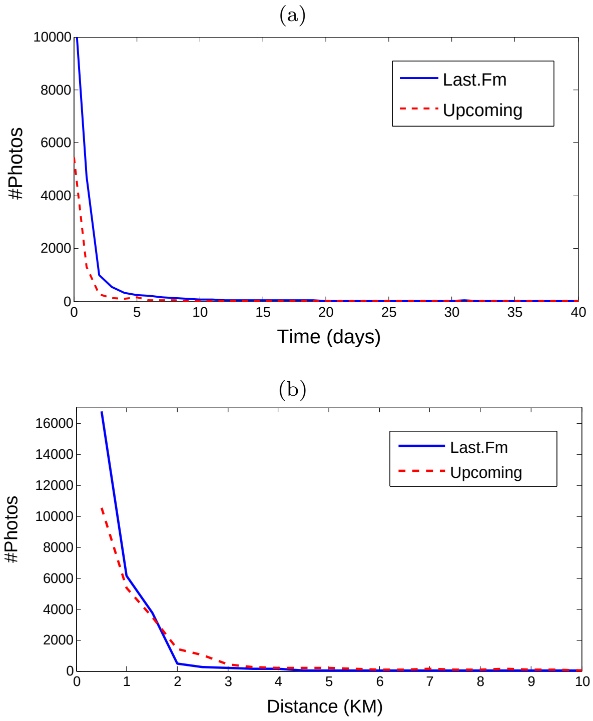


Fig. 5. The tendency to share photos around (a) event time and (b) event location

First, we investigate how people share photos considering the spatio-temporal and the topical dimensions. Figure 5 shows the tendency to upload photos with respect to the event start time and location. It can be gleaned that most of people upload photos right after the event started, and nearby the venue in which it took place. This let us suggest that one potential solution to identify some events is to detect the peaks of users activities within a narrow spatio-temporal window. In addition, we investigate how people upload photos according to attendance rate. Figure 6 highlights a strong correlation between the attendance rate and the amount of media shared. Moreover, we found out that most of active users are in general located in "United States" and "United Kingdom".

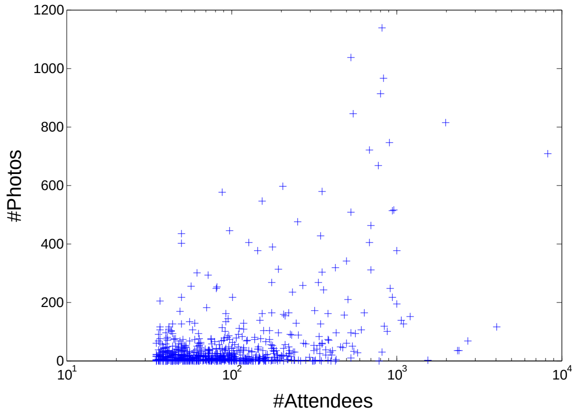


Fig. 6. The tendency to share photos according to the attendance rate

### 6.2. *How Many Attendees Involved?*

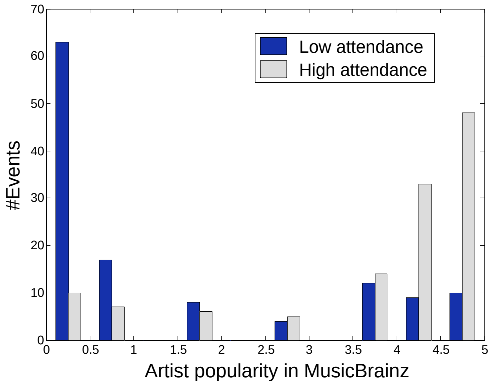


Fig. 7. Correlation between attendance rate and artist popularity

The second question we would like to address is: what are the events that prompt people to participate? Taking into account the high attendance rate, we distinguish two kinds of events. The first one encompasses the events featuring a significant number of artists. The second kind surprisingly includes events associated with few artists. To draw insight into attendance behavior to such typical events, we invoke additional information about artist popularity from MusicBrainz. Results are depicted in Figure 7. We can clearly observe that the artist popularity is an influential and important factor to attract people attending events.

### 6.3. *User Interest Model*

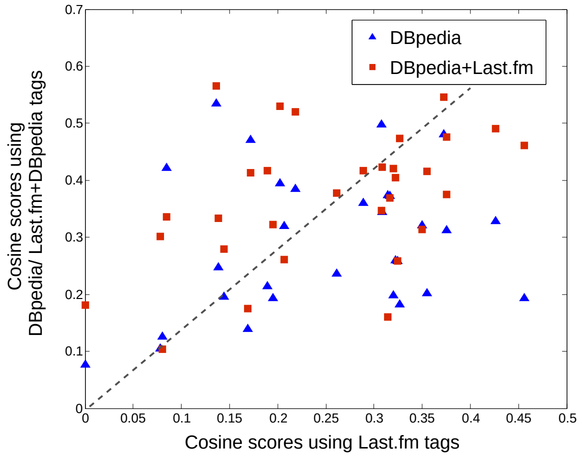
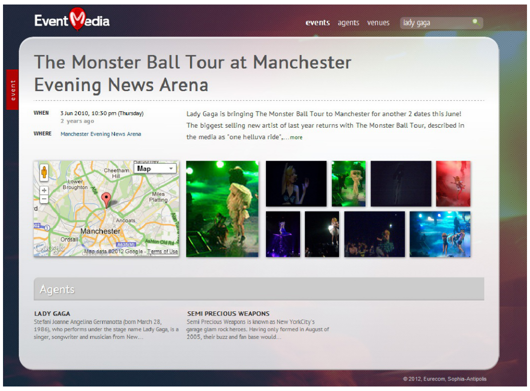


Fig. 8. User Interest Model with T = 10: comparison of Last.fm with DBpedia

Constructing a user profile is a crucial task in many recommender systems. To achieve this, one solution is to quantify the user interests using Latent Dirichlet Allocation (LDA) [1], a topic modeling technique. Inspired by the method in [15], the user interest model is reflected by the semantic annotations of items with which he has interacted. In this analysis, we aim to construct a user interest model comparing Last.fm annotations with DBpedia ones. Since an artist in general refers to one topic (music genre), we consider artists, of which the user has attended their shows, as the ideal items to detect the co-occurrence of tags. For each artist $i$, LDA generates a T-dimensional vector of topic proportions $\Theta_i = [\theta_i^1, \theta_i^2, ...\theta_i^T]$, where T is the number of topics. Then, we compute the variance of each topic dimension $t$ of all the artists $A$: $\Theta^t = [\theta_1^t, \theta_2^t, ...\theta_A^t]$. The user interest scores over T topics are represented by the vector $[\Theta^1, \Theta^2 ... \Theta^T]$. For this experiment, we retained only 39 users (among 400 users) that explicitly express their interests by means of tag-count pairs on their Last.fm home pages. For each user, we run several steps: (1) we retrieve the interesting tags (high count and topical tag) from the Last.fm home page, thus building a ground truth; (2) we collect the tags of appropriate artists from Last.fm (`dc:subject`) and DBpedia (`dbpedia-owl:genre`) on which we apply LDA; (3) We retain the tags of topics associated with high interest scores. Finally, we compute the Cosine similarity between the ground truth and the user interest model generated respectively using Last.fm, DBpedia and a combination of them. Figure 8 depicts a scatter plot of Cosine similarity scores. We can distinctly observe that DBpedia enhances the user interest modeling with the integration of more coherent and qualitative data.

## 7. Visualizing and Browsing Events

One challenge we want to address is how to enable fluid faceted navigation of a vast event-based space, and to create harmonious views of interconnected datasets. Users wish to discover events either through invitations and recommendations, or by filtering available events according to their interests and constraints. We provide mechanisms to browse events by location or a period of time. Once an event is selected, media are presented to convey the event experience, along with background information such as category, agents, venues, attendance list, ticket, etc. A typical example is illustrated in Figure 9.



The Monster Ball Tour at Manchester Evening News Arena

WHEN 3 Jun 2010, 10:30 pm (Thursday)
2 years ago

WHERE Manchester Evening News Arena

Lady Gaga is bringing The Monster Ball Tour to Manchester for another 2 dates this June! The biggest selling new artist of last year returns with The Monster Ball Tour, described in the media as "one helluva ride",...more

LADY GAGA
Stefani Joanne Angelina Germanotta (born March 28, 1986), who performs under the stage name Lady Gaga, is a singer, songwriter and musician from New...

SEMI PRECIOUS WEAPONS
Semi Precious Weapons is known as New YorkCity's garage glam rock heroes. Having only formed in August of 2005, their buzz and fan base would...

Fig. 9. Interface illustrating a concert of *Lady Gaga* in 2010

Apart from the inspection of the event instance, other conceptual classes (e.g. venues, agents, users) have also accessible views, so that the user can obtain more information about these instances and explore events related to them. We also leverage data from open datasets to be displayed in infobox separated from the main information, but some parts of this data are interwoven with the main data as well, or used to replace missing data. Finally, we enable the user to filter data by his favorite language. The demonstration of EventMedia is available at `http:`

//eventmedia.eurecom.fr. The reader is invited to watch http://eventmedia.eurecom.fr/demo.html before experimenting the live demo. From a technical point of view, we have been based on Elda, a java implementation that enables a configurable way to access RDF data using simple RESTful URLs that are translated into queries to a SPARQL endpoint. It provides a simplified XML and JSON representations of RDF data, suitable for use in the context of JavaScript Frameworks. We used a popular Backbone.js JavaScript framework [5] to facilitate developing the complex user interface. It is a simple but powerful MVC framework, providing Model, Collection, View and Router constructor, together with Event constructor for supporting Pub/Sub pattern. Moreover, it provides an elegant REST integration that makes dealing with Elda REST implementation straightforward.

## 8. Conclusion

Social events and media services host an ever increasing amount of knowledge, but spread and locked into multiple sites. Mining connections in this knowledge space is a key asset to enhance information exploration within a single channel. The work presented here falls within this perspective with a focus on Semantic Web technologies as powerful means to link data. We built EventMedia, an open dataset encapsulating the description of events and media, continuously synchronized with recent updates and reconciled in real-time with Linked Data. Finally, we highlight the benefits of Semantic Web to efficiently handle the analysis of sociological aspects. For future work, we would like to test our instance matching approach on other datasets, and test the decision tree to assess the role of low correlated properties [9]. In addition, we plan to conduct a comprehensive analysis of co-attendance offline social network.

## Acknowledgments

This work was supported by the project AAL-2009-2-049 “Adaptable Ambient Living Assistant” (ALIAS) and the project “Social Annotations and eXperiences” (S-MAX).

[5] http://backbonejs.org/

## References

[1] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[2] R. C. Eberhart, Y. Shi, Particle swarm optimization: developments, applications and resources, in: IEEE Congress on Evolutionary Computation, vol. 1, 2001.

[3] A. Fialho, R. Troncy, L. Hardman, C. Saathoff, A. Scherp, What’s on this evening? Designing User Support for Event-based Annotation and Exploration of Media, in: $1^{st}$ International Workshop on EVENTS - Recognising and tracking events on the Web and in real life, Athens, Greece, 2010.

[4] A. Jentzsch, R. Isele, C. Bizer, Silk - Generating RDF Links while publishing or consuming Linked Data, in: $9^{th}$ International Semantic Web Conference (ISWC’10), Shanghai, China, 2010.

[5] J. Kennedy, R. C. Eberhart, Particle swarm optimization, in: the IEEE International Conference on Neural Networks, vol. 4, 1995.

[6] H. Khrouf, R. Troncy, Eventmedia live: Reconciliating events descriptions in the web of data, in: $6^{th}$ International Workshop on Ontology Matching (OM’11), Bonn, Germany, 2011.

[7] H. Khrouf, R. Troncy, Eventmedia: a lod dataset of events illustrated with media, Semantic Web Journal, Special Issue on Linked Dataset descriptions, 2012. IOS Press, ISSN: 1570-0844.

[8] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, Soviet Physics Doklady 10 (1966) 707–710.

[9] K. Nguyen, R. Ichise, H.-B. Le, Learning approach for domain-independent linked data instance matching, in: $2^{nd}$ SIGKDD Workshop on Mining Data Semantics, Beijing, China, 2012.

[10] A. Nikolov, M. d’Aquin, E. Motta, Unsupervised learning of link discovery configuration, in: $9^{th}$ Extended Semantic Web Conference, Heraklion, Crete, Greece, 2012.

[11] X. Niu, S. Rong, Y. Zhang, H. Wang, Zhishi.links results for oaei 2011, in: $6^{th}$ International Workshop on Ontology Matching, Bonn, Germany, 2011.

[12] R. Shaw, R. Troncy, L. Hardman, LODE: Linking Open Descriptions Of Events, in: $4^{th}$ Asian Semantic Web Conference (ASWC’09), Shanghai, China, 2009.

[13] D. Song, J. Heflin, Automatically generating data linkages using a domain-independent candidate selection approach, in: $10^{th}$ International Semantic Web Conference (ISWC’11), Bonn, Germany, 2011.

[14] R. Troncy, A. T. S. Fialho, L. Hardman, C. Saathoff, Experiencing events through user-generated media, in: $1^{st}$ International Workshop on Consuming Linked Data, Shanghai, China, 2010.

[15] H. Wu, V. Sorathia, V. Prasanna, When diversity meets speciality: Friend recommendation in online social networks, ASE Human Journal 1.