# Enriching Live Event Participation with Social Network Content Analysis and Visualization

Marco Brambilla, Daniele Dell'Aglio(✉), Emanuele Della Valle, Andrea Mauri, and Riccardo Volonterio

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico of Milano, P.za L. Da Vinci, 32, 20133 Milano, Italy
{marco.brambilla,daniele.dellaglio,emanuele.della.valle, andrea.mauri,riccardo.volonterio}@polimi.it

**Abstract.** During live events like conferences or exhibitions, people nowadays share their opinions, multimedia contents, suggestions, related materials, and reports through social networking platforms, such as Twitter. However, live events also feature inherent complexity, in the sense that they comprise multiple parallel sessions or happenings (e.g., in a conference you have several sessions in different rooms). The focus of this research is to improve the experience of (local or remote) attendees, by exploiting the contents shared on the social networks. The framework gathers in real time the tweets related to the event, analyses them and links them to the specific sub-events they refer to. Attendees have an holistic view on what is happening and where, so as to get help when deciding what sub-event to attend. To achieve its goal, the application consumes data from different data sources: Twitter, the official event schedule, plus domain specific content (for instance, in case of a computer science conference, DBLP and Google Scholar). Such data is analyzed through a combination of semantic web, crowdsourcing (e.g., by soliciting further inputs from attendees), and machine learning techniques (including NLP and NER) for building a rich content base for the event. The paradigm is shown at work on a Computer Science conference (WWW 2013)

## 1 Introduction

During live events like conferences, exhibitions, and sports or fashion happenings, it has become common practice to share opinions, recommendations, materials, and reports through social media. Usually, the shared content refers to specific occurrences or objects related to the event, such as talks, speakers, exhibition stands, discussions, and so on. However, the mapping to such elements is often shallow or partial. This makes the social networking content an input not so valuable for the audience, especially if the social stream is very crowded and thus one has to deal with a big information overloading problem.

The problem tackled by this work is to enrich and classify the social media content related to a live event, in a way that makes it valuable for (local or


V. Presutti et al. (Eds.): ESWC Satellite Events 2014, LNCS 8798, pp. 159–170, 2014.
DOI: 10.1007/978-3-319-11955-7_13

remote) attendees. In particular, we focus on determining which contents are associated to which sub-event, and on enriching those contents with links to relevant entities (speakers, sessions, papers, and so on) in a domain-specific knowledge base. We then provide appropriate visualization to the enriched content, in a way that makes people able to understand what are the hot topics or sub-events and thus get guidance on what to do while attending the event.

In our approach, we select Twitter as the main social source for event-specific content. Twitter is indeed one of the most adopted platforms for social sharing, especially in the context of professional events: it can easily reach a large amount of interested people, messages are very short and require only few seconds to be shared. Furthermore, typically participants share their thoughts through event-specific hashtags, which are more or less officially related to the event itself, which makes it easy to associate them to the event.

We implement our solution in framework called ECSTASYS (Event-Centered STream Analysis SYStem) which combines semantic web, crowdsourcing (e.g., by soliciting further inputs by the attendees through social network invitations), natural language processing, named entity recognition and machine learning techniques for building a rich content base for the event. The application works in real time, processing the tweets as soon as they are available: in this way, attendees can have an updated and holistic view on what is happening and where, so as to get help when deciding what sub-event to attend. The application consumes data from different data sources: in addition to the afore mentioned Twitter, inputs include the official event schedule, plus domain specific content (for instance, in case of a computer science conference, DBLP and Google Scholar). The data processing determines the relevant entities described in the tweets and, consequently, the sub-events they relate to. The result of the analysis is shown to the attendees by room/sub-event, thus highlighting the interest and engagement of each sub-event, by means of appropriate user interfaces. The work is validated against a set of past conferences in the computer science field (for instance the WWW conference).

The paper is organized as follows: Sect. 2 gives an holistic view of the proposed solution, describing how the micro posts are processed. Sections 3 and 4 describe in detail respectively the data sources and the components that perform the data processing. Finally, Sect. 5 closes with possible future extensions.

## 2 The ECSTASYS Processing Flow

This section delves into the processing flow of the ECSTASYS framework by a logical point of view. ECSTASYS aims at augmenting the participation experience to live events through social network content enrichment and linking. To illustrate the processing flow, and to have a running example to use along the paper, we consider an experiment we conducted using ECSTASYS and depicted in Fig. 1: the scenario is the one of scientific conferences in the computer science domain, and in particular, the World Wide Web conference (WWW) 2013[1]. Conferences

[1] Cf. http://www2013.org/

are interesting complex events, with several parallel sub-events located in different rooms, typically in the same building. It follows that the precision error in geo-location would let infer wrong associations between tweets and sub-events. Moreover, people could discuss what happens in other rooms, so the geo-location is not enough to create the correct links. Demo and videos are available at http://demo.search-computing.com/aimc-2014/home.

During computer science conferences, as the WWW, a high number of participants uses social networks, and in particular on Twitter, to post messages describing the conference, e.g., updates on the talks, considerations on the keynotes, positive and negative opinions on the sessions they were in. The first step of ECSTASYS (Fig. 1a) is the retrieval of those posts: Twitter offers an API to retrieve in real time the tweets according to different criteria (e.g., location and keywords).

In this step, it is important to capture the highest number of relevant tweets, and this is paid in precision: ECSTASYS retrieves more tweets than required, gathering also non-relevant tweets, e.g., tweets talking about the quality of the dishes at the social dinner. The precision issue is addressed in the second step, the filtering (Fig. 1b): messages that are non-relevant for ECSTASYS are detected and discarded. ECSTASYS is built in this way due to the fact that the Twitter APIs have limited support for complex criteria definition, and consequently is hard to assess both high precision and recall in the first step.

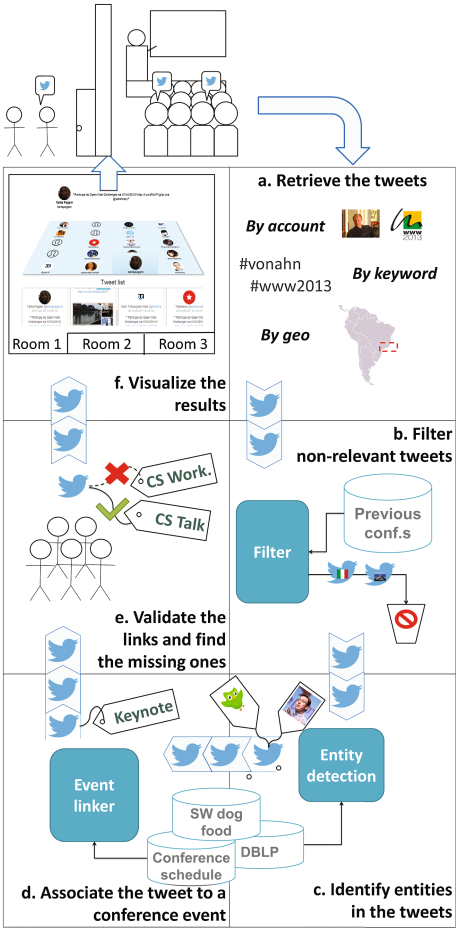


**Fig. 1.** ECSTASYS processing flow

In the third step (Fig. 1c), the WWW-related concepts mentioned in the tweet messages are identified. Attendants, presentations and rooms are example of concepts that ECSTASYS aims at detecting. In fact, they are keys to determine the conference events the tweets refer to: a tweet that discusses an event usually cites the presenter, the article or the room.

The identification of those entities in the text is at the basis of the following step, where ECSTASYS infers the links between tweets and events. This task

is performed in a semi-automated process: ECSTASYS makes an attempt to find the link automatically (Fig. 1d), and there are three possibilities: (1) a link is found with a confidence higher than a threshold; (2) there are more than one links with a high confidence; and (3) no links are found. In cases 2 and 3, additional manual tasks are executed, respectively a disambiguation and a link identification tasks (Fig. 1e). Finally, the tweets are visualised in a ad-hoc visualization that puts the emphasis on the associated event (Fig. 1f).

In the next sections, we go depth into the technical details of the ECSTASYS framework, explaining how the system implements the processing flow described above. First, we present the data, listing the sources and the componets we use to store and manage it; next, we discuss how it is processed.

## 3 Data Sources

ECSTASYS works with both dynamic and static data. Messages from Twitter are a typical example of dynamic data: a stream of time stamped messages updated at a high frequency. Additionally, as we see in Sect. 4, ECSTASYS requires static data to work, such as the description of the conference events and of the participants; this data is stored in a knowledge base, enriched with statistics such as term frequencies to improve the entity identification process.

### 3.1 Twitter

Twitter is the starting point of the whole approach: social feeds are retrieved by querying the Twitter Streaming API[2] based on hashtags, keywords, geographical locations, and people relevant to the event.

In the WWW experiment, We collected the tweets based on the hashtags of the conference (e.g., #WWW2013, #WWW, #vonahn), the location (i.e., the area around the conference building), and Twitter accounts related to the conference (e.g., the official twitter account – @www2013rio and Tim Berners-Lee – @timberners_lee). With these criteria, we collected more than 5000 tweets.

### 3.2 Domain Knowledge Base

The ECSTASYS knowledge base is the location on where the relevant data processed by the ECSTASYS components is stored. The knowledge base is exposed as a SPARQL endpoint and is built on the top of OpenRDF Sesame framework; as repository, we use OWLIM-Lite with the OWL 2 RL profile.

In our experiment, the knowledge base has been populated by: reusing some conference ontologies; importing the official data of the conference of interest; and importing bibliographic information about the people involved in the conference. We now report on this three aspects.

[2] Cf. https://dev.twitter.com/docs/api/streaming

**Ontology.** To design the ontology for ECSTASYS knowledge base, we reused existing ontologies: *(i)* the *Semantic Web Conference Ontology*[3], currently used to describe the data stored in the Semantic Web Dog Food repository[4] and describing conferences, related sub-events (e.g., keynotes, workshops, tutorials), talks and involved people with the different roles; and *(ii)* the BOTTARI ontology [5] for describing the tweets, an extension of the SIOC vocabulary to take into account the Twitter concepts (e.g., retweets, followers and followings). We also defined a set of custom concepts and properties to model the data produced by the ECSTASYS components that has to be stored: the mentions in the tweets, their relation with the entities and, consequently, the relations between the tweets and the events they relate to.

**Conference Data.** To describe the specific conference, we crawled the relevant information from the official Web site[5] and we performed the lifting from HTML/XML to RDF through XSPARQL [2] (information about the WWW 2013 conference is not available as linked data). This task required some manual work for setting up the crawler: in terms of effort, we spent one person day.

**Bibliography.** We use DBLP to enrich the ECSTASYS knowledge base with bibliographic information. We retrieved the list of the most recent papers written by each person involved in the conference (not only the authors, but also keynote speakers, organizers and chairs). As we describe in Sect. 4.4, this allows to enrich the keywords associated to each author and improve the precision of the entity detection step.

### 3.3 Domain Analytics

Analytics on the domain of interest are collected based on frequency of terms found in the social stream and of entities in the knowledge base. This aspect is important for reducing the impact of very frequent terms in the selected domain, which would not be considered as stop words in general sense but would actually generate noise in the specific domain. For instance, in our scenario terms such as *framework*, *solution*, *Web* are too frequent and their weight in the computation process is consequently lowered.

### 3.4 Crowd

The crowd is the source of input from human agents solicited by ECSTASYS. Typical collected information comprises confirmation of relevance of some entities for a tweet and selection of entities not automatically identified. In this case the crowd is compose by experts, since we target people attending the specific event. ECSTASYS builds the expert crowd by involving relevant tweets authors that are participating at the event.

[3] Cf. http://data.semanticweb.org/ns/swc/ontology
[4] Cf. http://data.semanticweb.org/
[5] Cf. http://www2013.org/

## 4 Processing Components

ECSTASYS puts together different techniques and tools to process and enrich the tweets. Figure 2 gives an overview of the framework, highlighting the data sources we presented above, and the components that implements the ECSTASYS processing flow. In the following, we present those components, explaining the technologies they use and how they work.

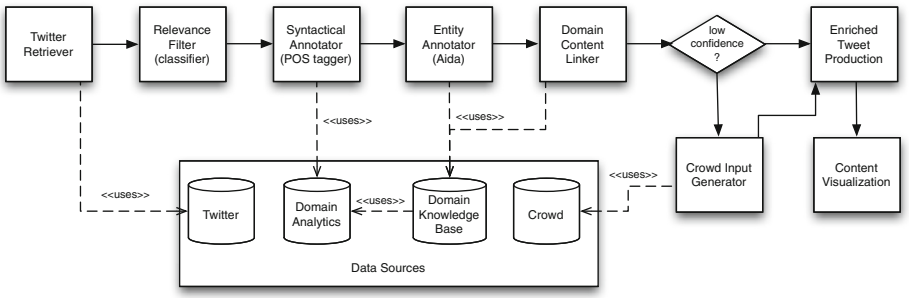


**Fig. 2.** Components of the ECSTASYS framework.

### 4.1 Twitter Retriever

The Twitter Retriever is the component that gathers the tweets that are relevant for the current event from Twitter. It uses the Twitter Stream APIs in order to connect itself to the public stream of tweets. This API allows to follow streams that match different predicates such as: users, keywords and location. All of these aspects are relevant for real world events, as they are typically identifiable by official hashtags, relevant people involved, and geographical coordinates of the venue.

### 4.2 Relevance Filter

The purpose of the Relevance Filter component is to filter out the non-relevant tweets that have been extracted by the Twitter Retriever but do not provide valuable information on the event. Typical examples include: tweets written in non-English language, tweets emitted in the prescribed geographical area or containing relevant keywords but not pertaining to the event, and so on.

The component immediately discards the tweets not written in English by looking at the *lang* field provided by Twitter as part of the tweet data structure. Furthermore, for selecting the relevant tweets we apply a classification approach, by exploiting a classifier based on Conditional Random Fields [10] trained on datasets coming from past events similar to the considered one. In particular we

built a wrapper of the *CRF++* implementation[6] in NodeJS, and we released it as open-source project on GitHub[7].

### 4.3 Syntactical Annotator

Once the relevant tweets are selected, they are annotated through a Part Of Speech (POS) tagger. The component provides as output the annotated tweet, plus a customized set of syntactical elements extracted from the text which will be useful for the extraction of entities. Such elements consist in set of words that are good candidates for becoming named entities. On this we propose a set of heuristic solutions aimed at increasing the recall of candidate terms for the extraction of entities, as opposed to classical off-the-shelf Named Entity Extractors, which feature very high precision but also limited recall. Some examples of heuristics we apply include: generation of all the possible aggregation of contiguous nouns, contiguous nouns and adjectives, and so on.
For instance the tweet "Ingenious way to learn languages: duolingo #keynote #www2013" is tagged in the following way:

> Ingenious‘JJ way‘NN to‘TO learn‘VB languages:‘NN duolingo‘NN #keynote‘NN #www‘NN 2013‘CD

The two-letter annotations (e.g., NN, JJ) are the POS tags, which define the grammatical role of a word inside the sentence. For instance, *NN* indicates that the word is a noun, *VB* indicates a verb, and so on.

Finally the application of the heuristic algorithm on the list of nouns produces the following aggregation: ["way"]["languages","duolingo"]. The nouns "keynote" and "www" are not considered because they are too frequent by the Domain Analytics. Our preliminary evaluation shows that the trained classifier achieves 81 % precision and 97 % recall when applied to the WWW 2012 content and 71 % precision and 84 % recall when applied to the WWW 2013 content.

### 4.4 Entity Annotator

The Entity Annotator component processes the data produced by the Syntactical Annotator so as to determine which are the entities discussed in the text. Among the existing named entity recognition (NER) tools, we selected one based on the following requirements:

- capability of performing real-time processing of content;
- capability of linking the text items to entities in an ontology. In the recent years, several entity annotators were built on the top of open data and public knowledge bases (e.g., DBpedia and freebase) [6,7].
- support of customization of the reference knowledge base to be used by the tool.

[6] Cf. http://crfpp.googlecode.com/svn/trunk/doc/index.html
[7] Cf. https://github.com/janez87/node-crf

The last requirement is extremely critical in our setting because usually entity annotators are only able to process generic textual content and to extract the generic entities (e.g., entities described in Wikipedia). However, in our case every event typically focuses on a very specific setting or domain, for which generic knowledge bases would contain only generic terms and very famous entities, while they would miss most of the less famous people and subjects. As an example, Dr. Jong-Deok Choi, keynote speaker at the WWW 2014[8], does not have a page on Wikipedia (and consequently, does not appear in DBpedia).

To cope with those requirements, we decided to use AIDA [9], an open-source entity detector developed at the Max Planck Institute. It takes as input a text, it detects the set of mentions, i.e., relevant portions of the text, and associates each of them to an entity. To do it, it exploits an internal entity base and it performs two kinds of analyses: on the one hand, it selects the set of potential candidate entities for each mention; on the other hand, it performs entity-to-entity analysis to determine the coherence among the candidates. The default entity base of AIDA is built on the top of YAGO [8], but it can be customised (or replaced) with another one. To fit our needs, we built a custom entity base tailored on the domain specific knowledge base of the experimental scenario (as explained in Sect. 3.2).

The custom version of AIDA is wrapped in the Content Linker component: it takes as input a tweet, and enriches it with a set of couples ($mention - entity$). The resulting tweet is pushed to the Domain Content Linker. Continuing the example introduced above, one of the mentions identified by the Syntactical Annotator is *Duolingo*; when the Entity Annotator processes the tweet, it associates the mention with the paper "Duolingo: learn a language for free while helping to translate the web" of Luis Von Ahn at the IUI 2013. This annotation is an example of entity detection enabled by DBLP: the enrichment of the AIDA entity base with the list of recent papers of the people involved in the conference increase the probability to discover them.

### 4.5 Domain Content Linker

The Domain Content Linker aims at creating the relations between the tweets and the specific sub-events of the event, extracted from the official conference program (e.g., workshops, talks, sessions). As input, the component receives the tweets annotated by the Entity Annotator, i.e., a tweet with a list of related entities; as output, it enriches the tweets with the URI of the event it relates to.

This component infers two different relations: *discusses*, that indicates that a tweet talks about one of the sub-events (independently on the temporal relation between the two, i.e., the tweet could be talking about something that happened in the past or that will happen in the future); and *discusses during*, a sub-relation that states that the tweet talks about a sub-event while it is ongoing. This distinction is important for visualization purposes.

[8] Cf. http://www2014.kr/

The linkage among the tweets and the events is performed in two steps. First, the Linker retrieves the candidate events: this is done by combining the entities in the AIDA entity base that annotate the tweet, with the information in the ECSTASYS domain knowledge base. We encoded the rules that determine the candidates as continuous SPARQL queries [1] that are executed by the C-SPARQL engine; ECSTASYS runs a lifting operation on the tweet stream (from JSON to RDF) to process it. For example, let's consider the query $q$: select the *events* in which the creator of the work $w$ is a participant, and $w$ is an annotation of the tweet $t$. Let the input $i$ be: "Ingenious way to learn languages: duolingo #keynote #www2013 #gwap", annotated with the mention–entity: (Duolingo, "Duolingo: learn a language for free while helping to translate the web"). The evaluation of $q$ over $i$ produces the list of events in which Luis Von Ahn participates.

If a tweet has more than one annotation, the first step produces a set of candidate events; the second step works on it in order to derive an ordered list of candidates, associating to each of them a confidence value. The score is determined by the number of repetitions of the events in the multiset, and by their *temporal distance* to the tweet, i.e., it is more probable that a tweet discusses an event occurring temporally near. For instance, among the events on which Luis Ahn participated at the WWW 2013, the tweet was posted during the keynote, so it is the event with the highest rank in the output.

The time stamp of the tweet and the event scheduled time are also used to determine if the event can be related to the tweet through a *discusses during* relation: if the tweet is posted within 30 min before/after the event, the *discusses during* relation can hold.

### 4.6 Crowd Input Generator

The Crowd Input Generator use crowd-sourcing techniques to improve the precision of the framework and solve the ambigous results of the Domain Content Linker. This component is based on the CrowdSearcher framework [3,4], which allows planning and control of crowdsourcing campaigns. The component is triggered by specific events (e.g., tweets that cannot be associated with any sub-event, or tweets for which the confidence of the association is low), and assigns them to the crowd for getting feedback. The invitation to respond is sent to people relevant to the event (e.g., the author of the tweet himself and people who twitted about the event).

Figure 3 shows the Web interface used by the crowd in order to provide the answer to the crowdsourcing tasks proposed by ECSTASYS. On the top of the form is shown one tweet, with a list of possible choices (determined by the Domain Content Linker). The crowd user has to choose the correct room/session from the list. Due to the fact that ECSTASYS is using an expert crowd (i.e., people attending the event), it considers a tweet as evaluated as soon as one performer provides an answer. In order to involve as many participants as possible ECSTASYS exposes a user interface compatible with both mobile and desktop devices.

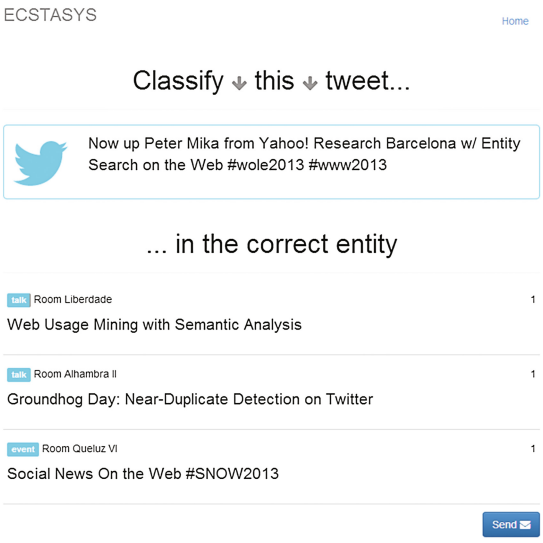


**Fig. 3.** Interface used by the crowd for providing the answers

### 4.7 Enriched Tweet Production

The Enriched Tweet Production component is in charge to create the tweet with the additional metadata generated by the ECSTASYS components, e.g., the information from the crowd, from the Domain Content Linker and so on. The output of the component is a stream of JSon tweets with additional ECSTASYS-related fields. Those tweets are then visible through the Content Visualization component, but in general they can also be used for further processing.

**Listing 1.** Example of an enriched tweet

```
{
 "created_at":"Mon May 06 15:01:02 +0000 2013",
 "id" : 331423366131101700,
                                        "...."
  "text":"don't miss the first ever #WWW2013 Linked Media w/s Monday
      @www2013rio promoting #semanticmedia and #mediafragments http://t.co/
      CWanSYwnj8",
 "names" : [ [ [ "t" ] ], [ [ "first" ] ], [ [ "Media" ] ], [ [ "/" ] ], [ [ "
     Monday" ] ], [ [ "rio" ] ] ],
  "taggedText":"don`VB''\" t`NN miss`VB the`DT first`NN ever`RB #WWW`NN 2013`
      CD Linked`VBN Media`NNP w`IN /`NN s`PRP Monday`NNP @www`NN 2013`CD rio`
      NN promoting`VBG #semanticmedia`NN and`CC #mediafragments`NNS http://t.
      co/CWanSYwnj8`URL",
 "relevant": true,
 "tweetAnnotations": [ {
   "mention":"WWW2013 Linked Media w/s",
   "entity":"http://www2013.org/program/first-worldwide-web-workshop-on-linked
       -media-lime2013/"
  } ],
 "relatedEvents": [ {
   "entity":"http://www2013.org/program/first-worldwide-web-workshop-on-linked
       -media-lime2013/",
   "prob" : 1
  } ]
}
```

Listing 1 shows an example of an enriched tweet. In bold (from Line 6) are highlighted the fields that are added by ECSTASYS. To make some examples, the *relevant* field (Line 8) contains the result of the Relevance Filter: it is a boolean that indicates if a tweet is relevant. The *taggedText* (Line 7) and *names* (Line 6) are the results generated by the Syntactic Generator: the former is the tweet message annotated with the POS tags, while the latter is the array with the relevant nouns sequences. The *tweetAnnotations* field (Line 9) contains the result of the Entity Annotator: the value is an array with the mentions found in the text message and the relative associated entity. Finally, the *relatedEvents* field (Line 13) contains the events related to the tweets found by the Domain Content Linker and the Crowd Input Generator, with the relative probability.

### 4.8 Content Visualization

ECSTASYS provides two types of visualizations for the enriched stream of tweets, as shown in Fig. 4. Both of them are web applications written in HTML5 and Javascript.

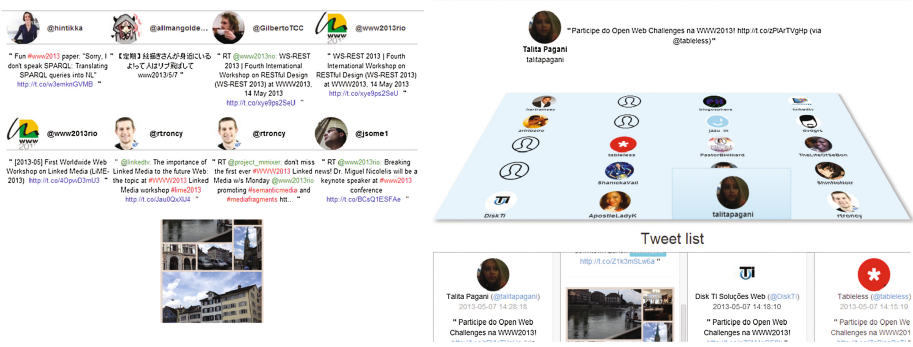


**Fig. 4.** *Wall* (left) and *Room* (right) visualizations of the enriched content.

The *Wall* visualization is meant to be used at the event venue on large panels (e.g., on screens or projectors in the lobby or outside the rooms of the sessions). It shows the tweets with highlighted author, mentions, hashtags and URLs. Rich media content linked by the tweets is shown separately at the bottom. The *Room* visualization instead aims at personal use (e.g., on desktop browsers) and mimics the layout of a room where a sub-event is happening. It shows a 3D view of the audience (i.e., people that twitted something related to the current sub-event) in the center, with the last relevant tweet on top. The author of the tweet flips up in the audience layout. At the bottom, a continuous slider shows the tweet stream. Each tweet appears with related media, highlighted URLs, mentions and hashtags.

## 5 Conclusions and Next Steps

In this paper we presented ECSTASYS, a framework for improving the experience of event attendees by exploiting and enriching the contents shared on the social networks. The prototype we developed prove the feasibility of the system, but additional work is required. First, we need to evaluate it: at the moment we have just some indicators about some components (e.g., the relevance filter), but we plan to evaluate the precision and recall of each separate components first, and of the whole system then. Additionally, we aims at improving the ECSTASYS components, e.g., design a more sophisticate heuristic algorithm for the extraction of syntactical elements and develop more precise crowd activation and control rules. Finally, we will investigate the generality of approach, deploying ECSTASYS during other conferences and more in general other kinds of events.

## References

1. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-sparql: a continuous query language for rdf data streams. Int. J. Semant. Comput. **4**(1), 3–25 (2010)
2. Bischof, S., Decker, S., Krennwallner, T., Lopes, N., Polleres, A.: Mapping between rdf and xml with xsparql. J. Data Semant. **1**(3), 147–185 (2012)
3. Bozzon, A., Brambilla, M., Ceri, S.: Answering search queries with crowdsearcher. In: 21st World Wide Web Conference (WWW 2012), pp. 1009–1018 (2012)
4. Bozzon, A., Brambilla, M., Ceri, S., Mauri, A.: Reactive crowdsourcing. In: 22nd World Wide Web Conference, WWW '13, pp. 153–164 (2013)
5. Celino, I., Dell'Aglio, D., Della Valle, E., Huang, Y., Lee, T., Kim, S.-H., Tresp, V.: Towards BOTTARI: using stream reasoning to make sense of location-based micro-posts. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) ESWC 2011. LNCS, vol. 7117, pp. 80–87. Springer, Heidelberg (2012)
6. Cornolti, M., Ferragina, P., Ciaramita, M.: A framework for benchmarking entity-annotation systems. In: Proceedings of the 22nd International Conference on World Wide Web, WWW '13, pp. 249–260 (2013)
7. Gangemi, A.: A comparison of knowledge extraction tools for the semantic web. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 351–366. Springer, Heidelberg (2013)
8. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: a spatially and temporally enhanced knowledge base from wikipedia. Artif. Intell. **194**, 28–61 (2013)
9. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP2011), pp. 782–792 (2011)
10. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML '01: Proceedings of the 18th International Conference on Machine Learning, pp. 282–289 (2001)