Enabling Self-Service Data Provisioning Through Semantic Enrichment of Data

Ahmad Assaf

A doctoral dissertation submitted to:

TELECOM ParisTech

in partial fulfillment of the requirements for the degree of:

Doctor of Philosophy

Specialty : Computer Science and Multimedia

Supervisor:
Dr. Raphaël Troncy - EURECOM, France

Acknowledgments

Working as a PhD student in Eurecom was a great experience that would not be achieved without the help and support of many people, who I would like to acknowledge here.

First and foremost, I would like to thank my supervisor Dr. Raphaël Troncy for his invaluable support and great guidance throughout my study. I would like to express my gratitude to him for provided me with a lot of freedom to pursue my research. This work would not have been possible without his scientific knowledge and constructive advice.

I would like to extend my sincere thanks to my committee members, the reviewers Prof. Geert-Jan Houben and Dr. Catherine Faron-Zucker, and the examiners Prof. Talel Abdessalem, Prof. John Domingue and Dr. Tommaso Di Noia for their precious time and shared insights.

I am particularly indebted to my colleagues who more or less directly contributed to my Ph.D. Precisely, I would like to acknowledge the support of my friends: Vuk, Ghislain, José Luis and Giuseppe for their inspiring collaboration and fruitful discussions. It was a pleasure to work and exchange with them. Also, I thank all those working at EURECOM, they made my stay very pleasant.

I owe my deepest gratitude to my parents, Nejiba and Abdessalem, my sisters Sonia and Hela and my brother Ahmed for their unwavering encouragement, devotion and love. Last but not least, special thanks go to my friends for their constant friendship, moral and infinite support.

Abstract

Recently, the widespread growth of social media has shifted the way people explore and share information of interest. Part of this evolution is the event landscape increasingly augmented by user-generated content leading to vast amount of event-centric data. In today's Web, numerous are the services that provide facilities to organize and publish events, and to share related thoughts and captured media. However, the information about the events, the social interactions and the representative media are all spread and locked into the sites providing limited event coverage and no interoperability of the description. To fully benefit from the event, users are constrained to monitor different channels, some of which suffer from the information overload problem.

The goal of this thesis is to provide a unified environment that provides broad event coverage along with complete description and illustrative media, and to investigate efficient approaches that can benefit content personalization. The major challenge is to face the complex nature of events as multifaceted, ephemeral and social entities.

Various distributed platforms host a wide variety of scheduled events along with related media and background knowledge, making the user-contributed Web a primary source of information about any real world happening. Mining in real-time the connections between these heterogeneous and spread data fragments is a key factor to improve data quality and to enable opportunistic discovery of events. Towards this goal, we integrate different sources using Linked Data, so that we can explore the information with the flexibility afforded by Semantic Web technologies. More precisely, we leverage the wealth of information derived from event-based services, media platforms and social networks to build a Web environment that allows users discovering meaningful connections between events, media and people.

On the other hand, users tend to be overwhelmed by the massive amount of information available in event-based services. This fact requires valuable solutions that cope with the information overload. In particular, recommendation and community detection are two promising solutions that have been widely investigated in research. Yet their applications in event domain are still elusive. Thus as a first solution, we propose a hybrid recommender system that capitalizes on ontology-based event representation along with the collaborative filtering techniques. Second, we propose an approach to discover topical communities in event-based social network. Both the network links and the event topics are examined during the clustering process in which the quality function is the so-called semantic modularity.

Contents

List of Figures

List of Tables

Glossary

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first appears in the text.

AI	Artificial Intelligence
API	Application Programming Interface
CB	Content Based recommender system
CDF	Cumulative Distribution Function
CF	Collaborative Filtering
EAV	EntityAttributeValue model
EBSN	Event-based Social Network
ELDA	Epimorphics Linked Data API
FOAF	Friend of a friend
GA	Genetic Algorithm
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IR	Information Retrieval
ISBN	International Standard Book Number
JSON	JavaScript Object Notation
LBSN	Location-based Social Network
LDA	Latent Dirichlet Allocation
LOD	Linked Open Data
LODE	Linking Open Descriptions of Events
NE	Named Entity
NER	Named Entity recognition
NERD	Named Entity Recognition and Disambiguation
NLP	Natural Language Processing
OWL	Web Ontology Language
PLSA	Probabilistic Latent Semantic Analysis
PSO	Particle Swarm Optimization
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
RSS	Really Simple Syndication
RSVP	Request for a Response
SIOC	Semantically-Interlinked Online Communities
SKOS	Simple Knowledge Organization System
SPARQL	Query Language for RDF
SUS	Stochastic Universal Sampling

TDT	Topic Detection and Tracking
URI	Universal Resource Identifier
URL	Universal Resource Locator
VSM	Vector Space Model
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Chapter 1

Introduction

Services such as event directories, social networks and media platforms host an ever increasing amount of event-centric data. Recently, they have attracted people to organize and distribute their personal data according to occurring events, to share related media and to create new social connections. Still, this data needs to be structured and integrated in order to enhance different tasks such as content presentation, recommendation and social analysis.

1.1 Context and Motivation

Roughly speaking, "*event*" is a phenomena that has happened or scheduled to happen at a specific place and time. According to recent studies in neuroscience [183], event is also considered as past experience with which humans remember their real life. A common practice for humans is to naturally organize their personal data according to occurring events: wedding, conference, concert, party, etc. They would like to plan activities according to future events or to record what happened during past events.

Along with the emergence of Web 2.0, people become more involved in online activities sharing rich content to describe events and engaging in social interactions. This is reflected in many social services where a large amount of data exists in multiple modalities such as event attributes (e.g. time, location) and explicit RSVP (i.e. expressing the user intent to join social events) in event directories (e.g. Eventful, Last.fm, Lanyrd, Facebook), photos and videos captured during events and shared on media platforms (e.g. Flickr, YouTube), digital chatter generated by reactions to events in social network sites (e.g. Twitter, Facebook). Yet, this knowledge forms a huge space of disconnected data fragments providing limited event coverage [54]. For instance, while Last.fm sustains a broad coverage on event attendance, other valuable details are often missing such as description, price and media. Users tend to use other channels to complement the event overview. Moreover, most of event directories provide limited browsing options (e.g. lack of location map) and unreliable event recommendation (e.g. no consideration of like-minded users). These limitations have been notably highlighted in an exploratory user centered study conducted to assess the perceived benefits and drawbacks of event websites [172]. Having in mind the findings of this study, we focus on two major tasks which are data reconciliation and personalization.

1.1.1 Data Reconciliation

A large amount of event-centric data is spread across multiple services, however, often incomplete and always locked into the sites. How to leverage the wealth of this information is a serious challenge towards providing a broad coverage of events. As a solution, integrating data is a prominent way to deliver more complete and accurate information. In particular, the recent use of Semantic Web technologies proves to be effective to ensure a large-scale and flexible data integration. With ontologies, developers can structure large amounts of heterogeneous data independently of particular applications, while explicitly representing enriched semantics. In order to deliver enriched event views and higher information coverage, two ways can be explored in data reconciliation with a focus on Semantic Web technologies.

The Semantic Web is predicated on the availability of large amounts of structured data as RDF, not in isolated islands, but as a Web of interlinked datasets. Linked Data[1] is an ongoing project pursuing this avenue and connecting related data through RDF triples[18]. It interlinks RDF datasets on a large scale and follows the principles[2] outlined by Tim Berners-Lee in 2006. A fundamental concern in the Semantic Web is the comparison and matching of structured data to achieve the vision of the Linked Data. However, data sources often do not share commonly accepted identifiers (i.e. ISBN codes) and make use of different vocabularies. In particular, data reconciliation has recently gained importance in the Semantic Web community and it comprises two main sub-tasks: the former is the ontology matching which refers to the process of determining correspondences between ontological concepts; the latter is the instance matching which refers to the process of determining correspondences between individuals. In this thesis, we focus on the instance matching task to discover identical individuals referring to the same real-world entity. Indeed, the availability of events provided by disparate Web services increases not only the amount of data, but also the variety of representations of a single event-centric entity (e.g. location, participant, artists, etc.). It has been shown that the reconciled data are more advantageous to enhance data quality by improving both completeness and accuracy [137]. For example, one data source may contain events with few details about involved artists. Another data source may complement the description by providing the biography with complete discography of those artists.

The second solution is to associate events with user-contributed social media. In fact, real-world events often trigger a tremendous activity on numerous social media platforms. Participants share captured photos and videos during events, tweet status messages and engage in discussions with comments. To mine the intrinsic relationships between events and media, most of existing studies focus on event detection from user-generated content that describes breaking news or social events [117, 9, 155]. Automatic event detection is essentially a clustering problem aiming to group together

[1] http://linkeddata.org/
[2] http://www.w3.org/DesignIssues/LinkedData.html

media documents discussing the same event. Other existing works study this problem within the field of data reconciliation [150, 50]. The idea behind is to compare instances of different ontological classes (e.g. event class and media class) using their related features such as named entities and contextual information. In this thesis, we exploit this idea and we bridge the gap between structured events and unstructured media data.

Reconciling event-centric entities or aligning events with media have in common some challenges that originate from the use of online, heterogeneous and distributed sources. First, the same real-world entity is often represented in different ways across the disparate data sources. Some of these entities may be related with short descriptions and featuring noisy information. Moreover, the user-generated content exists typically at large scale and evolves dynamically providing daily a significant amount of events, locations, media, etc. These challenges demand a scalable, real-time and efficient techniques to integrate data.

1.1.2 Personalization

Personalization in online social services have gained momentum over the recent past years. Providing assistance to make decision and select reliable products become part of primary concerns in the e-service area. More specifically, integrating personalization techniques in event-based services is a key advantage to attract people to attend relevant events. Such techniques recently start to draw attention as has been attested by the VP (Vice President) Operations of Eventful reporting that "*When we really got serious about personalization, we started talking about it a few years ago and we really got busy a couple of years ago*"[3].

One personalization technique is to build a recommender system that decodes the user interests and optimizes accordingly the information perceived. To help such system predict items of interest, various clues are available ranging from the user profile, explicit ratings, to past activities and social interactions. Different from a classic item, event occurs at a specific place and during a period of time to become worthless for recommendation. While the classic items continuously receive useful feedback, the user preferences related to events are very sparse. This problem is mainly due to the transient nature of events leading to the fact that most of users are associated with very limited number of events. Given this high sparsity level, traditional recommender systems fail to handle event recommendation where both content and social information need to be considered [36].

Another innovative technique is to position the user within one or more communities, instead of an isolated individual [140]. In order to enable community-driven personalization, the system needs to analyze networked data and reveal the underlying communities. This demands an efficient method to detect meaningful communities

[3]Paul Ramirez, MarketingSherpa Email Summit 2014.

which in turn can benefit various tasks such as customer segmentation, recommendation and influence analysis. In research, several studies has been devoted to solve this problem, but mostly focused on the linkage structure of the network. They assume that the proximity of users is solely reflected by their interactions strength. However, such methods do not consider the topical dimension and often group users having different interests. This problem becomes important when a user interacts with different social objects (e.g. events) inducing highly diverse topics in his/her profile. Consequently, there is a need to incorporate the semantic information along with the linkage structure for detecting meaningful and overlapping communities [38, 185].

In this thesis, we tackle the problems related to event recommendation and to community detection in event-based social network. The challenge is to deal with the complex nature of events where social and content information are both important.

1.2 Thesis Contributions

As a multidimensional, ephemeral and social entity, the notion of "*event*" poses new challenges for research community. In this thesis, we propose approaches related to data reconciliation and personalization in event domain. In summary, the main contributions of this work are as follows:

- We created a framework in order to aggregate in real-time event-centric data retrieved from heterogeneous sources. Our strategy is to build an architecture flexible enough to accommodate ongoing growth. Such flexibility is ensured by the ease to add new sources and the use of Semantic Web technologies. The data, continuously collected in real-time, is converted to RDF using existing vocabularies and then stored in a triple store. The entire dataset is called EventMedia.

- We propose heuristics to mine the intrinsic connections of event-centric data derived from event directories, media platforms and Linked Data. Given the dynamics of social services, our approach ensures a real-time reconciliation maintaining a dynamic content enhancement. First, we propose a domain-independent reconciliation approach that identifies identical entities residing at heterogeneous sources. Then, we tackle the problem of aligning structured events with unstructured media items based on Natural Language Processing (NLP) techniques.

- We consumed Linked Data in order to develop friendly Web applications that meet the user needs: relive experiences based on background knowledge and help create events with consistent details. Then, we highlight the benefits of Linked Data to steer the behavioral analysis and to improve the user profiling.

- We propose a hybrid system to recommend events based on content features and collaborative participation. This system enriches an event profile with Linked Data and exploits the ontology-enabled feature extraction. It is also enhanced by an approach that detects the effective user interests within a topically diverse user profile.
- We introduce a novel approach that detects topical communities within event-based social networks. We distinguish between online and offline networks constructed based on the collaborative participation in events. Our approach exploits the hierarchical clustering with the combination of both the content features and the the linkage structure. Then, a link-based function is defined to determine the effective user attachment to each community.

1.3 Thesis Outline

The work presented in this thesis first describes how to integrate event-centric data into a Semantic Web dataset. Then, it focuses on consuming Linked Data in event domain for the development of Web applications and personalization methods.

Chapter 2 is dedicated to overview the background of our work including the research in event domain and some paradigms related to Semantic Web. We first introduce the important aspects related to events and the basic concepts in the Semantic Web. Then, we describe the evaluation criterion used throughout this work. The rest of this manuscript is composed of two major parts:

1. In the first part, we focus on the building task that retrieves event-centric data from distributed sources and integrates them into one semantic knowledge base. Such task includes crawling, structuring and linking data, which needs to be ensured with the flexibility afforded by the Semantic Web technologies. The contributions of this part have been published in [95, 96, 93, 90].

    - **Chapter 4** describes how data is extracted, structured and published following the best practices of the Semantic Web. In particular, we pay attention to create a flexible framework that performs those tasks, and eases the addition of event and media Web services.
    - **Chapter ??** studies the problem of data reconciliation in a heterogeneous environment. We present our approach to detect identical entities in event-centric data by the use of instance matching techniques. Then, we propose a NER-based approach to align events with microposts, thus bridging the gap between structured and unstructured content.

2. In the second part, we exploit the constructed knowledge base for various applications. The goal is to highlight the benefits of linked data to improve the

event presentation and to explore solutions for personalization in event-based services. The contributions of this part have been published in [94, 92, 91, 97].

- **Chapter 5** presents three Web applications in charge to support better visualization and to help users search, browse and create events. Besides, it underlines the benefits of our knowledge base, as part of Linked Data, to understand some facts about the user behavior and to improve the user profiling.
- **Chapter 6** presents our approach built on top of Semantic Web to recommend social events. The idea is to leverage structured and expressive representation of events to predict what a user likes. Our approach is then augmented by the recommendation based on collaborative filtering.
- **Chapter 7** exploits event-centric users activities in order to construct event-based social networks in online and offline worlds. Then, we propose an approach to detect meaningful communities taking into account the event topics and the linkage structure of the network.

Chapter 8 concludes the presented work and outlines new research directions.

Chapter 2

Background

In the last few years, an increasing interest in event domain has led to diverse contributions in research. In this chapter, we provide a background analysis on the definition of "*event*" in the Social Web and on the perceived qualities of available event directories. Then, we overview the Semantic Web technologies considered as powerful means to ensure a large scale data integration. Finally, we present some evaluation metrics used throughout this thesis.

2.1 Events on the Web

An ever increasing amount of event-centric knowledge is spread over multiple social services, either materialized as calendar of events or illustrated by cross-media documents. Determining what an event is and how people use those services are two important research questions. In this section, we present the event definition adopted in this thesis, and we provide an overview of some social services as well as the perceived benefits and drawbacks of using them.

2.1.1 Event Definition

What is meant by the word "event"? has always been a research question leading to several meanings. This subject has received substantial consideration across different fields such as philosophy [29] and computer science [4]. From a broader point of view, a real event is considered as something that happens: a happening, an occurrence, an event [157]. This definition has been extended in a philosophical study to characterize events as an abstract concept in which the meaning depends on the target type such as activity, state or action [29]. From technical point of view, an earlier work in Topic Detection and Tracking (TDT) field defines an event "as something that happens at a particular time and place" [4]. This definition puts emphasis on the spatial-temporal aspect, which seems to be adopted by many other researchers [115, 184]. However, while events can happen at a specific time, other events continue over a long period of time. Moreover, associating a specific location to events fails to handle some events which may happen in different venues. These facts have led to other definitions in the literature attempting to cast an event to just a temporal entity [145] or to stress on the geographical dimension [162]. To sum up, by drawing together all these definitions, three important views appear to identify what an event is. These views are represented by three *Ws* questions: *what*, *when* and *where*.

Later on, some researchers point out a missing concept that could define an event. They attempt to pay attention to "*who*" was involved in the event. Although events can happen without participants, it seems important to consider this aspect when it comes to describe the people's experiences. Thus, the definition in [4] has been extended to "an event is something that has a specific time, location, and people associated with it" [3]. For instance, it has been shown that the "*who*" view is important to define a historical event which is described by five elements: object, person, location, time and cause [136]. While causality appear in some definitions, it is of less significance to our work since we are not primarily interested in linking events by cause/effect relationships.

In [160], the authors proposed a study to compare existing models that attempt to represent events in a structured format. They propose an interoperable model to represent intersubjective "consensus reality" over all event definitions. Based on this model, we define an event in terms of the four *Ws* questions as follows:

1. *What* happened: represented by a set of terms.
2. *Where* it happened: associates an event with any number of places.
3. *When* it happened: associates an event with a specific time or period of time.
4. *Who* was involved: distinguishes between people having "active" or "passive" role.

2.1.2 Social Websites

Events on the Web exist in two different types: *unstructured* and *structured*. Unstructured events are mostly represented in form of natural language phrases which require complex parsing and extraction mechanisms. On the other hand, structured events are represented in a well-defined structure that may differ from one site to another. Currently, there exists a large variety of websites that host structured information about past and upcoming events, some of which may display media. In this thesis, we focus on structured events as provided by some popular event websites. In the following, we provide an overview about these sites as well the platforms which host related media.

Event Sites

Many Web services available online aim to help users search and share information about past and upcoming events. Whilst some websites focus on a specific type of events (e.g. musical, conference), other ones provide a wide span of different types including film, theater, exhibition, etc. In this thesis, we use some popular event sites described as following.

- **Last.fm**[1]**:** is the largest music based platform founded in 2002 having more than 30 million active users. It allows to build a user profile based on listening preferences of music collection or radio station. In October 2006, Last.fm incorporated a system that lets users post musical concerts with some details (date, venue, location, artists, etc.). They are also able to express their intent to attend events using RSVP (e.g. *I'm going*). Tags and comments are also possible on almost any item such as a user, event, artist, or track. Finally, users can register in any group which may be linked to artists or countries, and can add other users as friends. Figure 2.1 depicts the homepage of Last.fm.

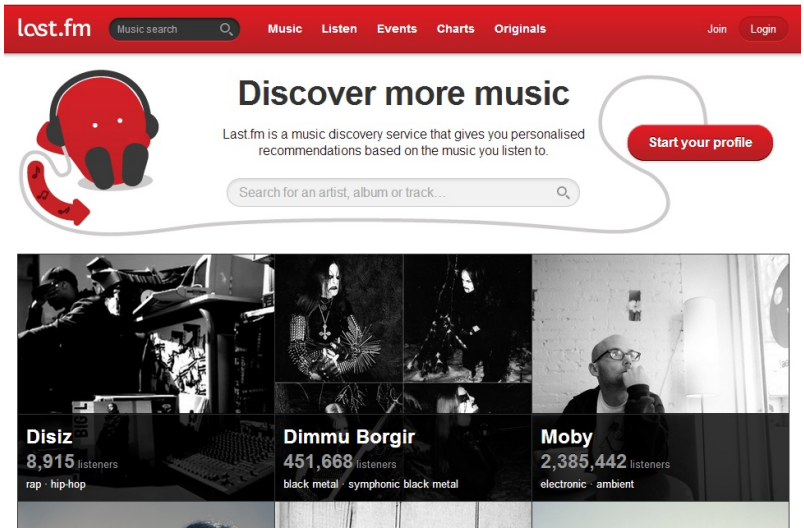

Figure 2.1: Last.fm

- **Eventful**[2]**:** is a popular event-based service founded in 2004. It boasts one of the world's largest databases of events having a wide variety of types such as sport, cinema, family, education and other local entertainment. It allows users searching for events by location, time, category, artist and descriptive keyword. It also provides functionality to view and manage a list of favorite artists and venues. Figure 2.2 depicts the homepage of Eventful.

- **Lanyrd**[3]**:** was founded in 2010 providing a social directory of conferences and other professional events. It enables users to enter location, speakers, schedule and other descriptive details. Users can be identified through the Twitter[4] or LinkedIn[5] API and are invited to list the conferences which they are attending or speaking at. Figure 2.3 depicts the homepage of Lanyrd.

[1] http://www.last.fm
[2] http://www.eventful.com
[3] http://www.lanyrd.com
[4] http://www.twitter.com
[5] http://www.linkedin.com

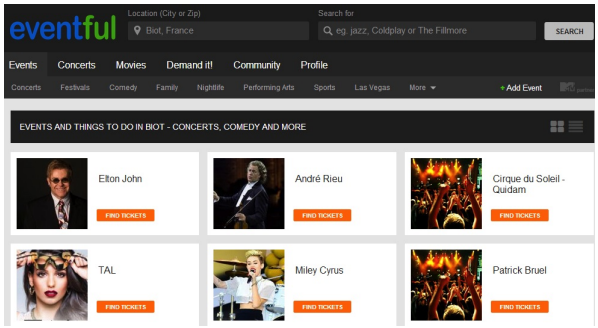

Figure 2.2: Eventful

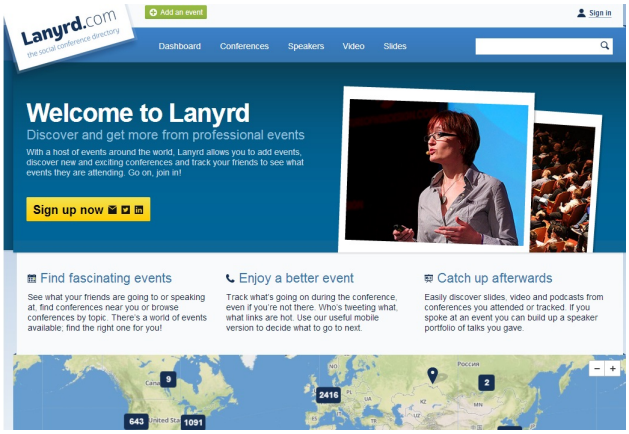

Figure 2.3: Lanyrd

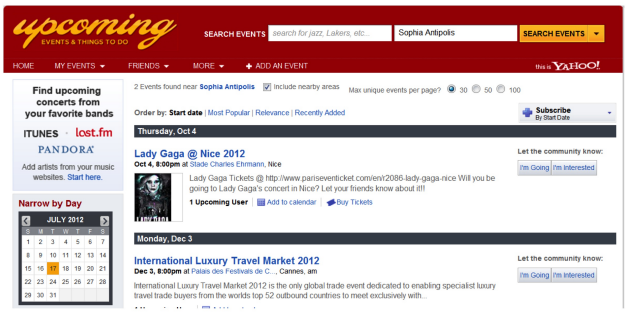

Figure 2.4: Upcoming

- **Upcoming:** was another event-based service launched in 2003, acquired by Yahoo! in 2005 but retired in 2013. It was a competitor of Eventful and offers similar functions. Upcoming hosted different types of events such as conferences, art exhibitions along with useful details including time, location, etc. Users can create and manage events and have a "friend" relationship with each other. Figure 2.4 depicts the homepage of Upcoming.

Media Sites

Many participants use social media platforms to engage in discussions with comments, and share media captured during events. In the following, we describe some popular media sites involved in this thesis.

- **Flickr[6]:** is one important photo and video sharing website founded in 2004. The site claimed 6 billion of hosted photos in 2012 witnessing a significant growth in the past few years. It provides rich data about photos that have been widely exploited by research community. This data describes several attributes such as title, description, uploading time, geo-coordinates, tags, etc. One popular attribute is the so-called "machine tag" or "triple tag" which is used in this thesis. It is based on a special syntax which is meaningful to be processed by machines. It comprises three parts: (1) the namespace to denote the classification of a tag ('flickr', 'geo', etc.); (2) the predicate to represent the property of the namespace; (3) and the value of the tag. For instance, "geo:lat=25.070173" is a tag for the geographical latitude whose value is 25.070173.

- **Twitter:** is the most famous micro-blogging service emerged in 2006. The site claimed 200 million of active users in February 2013. It allows users and organizations to publish text messages limited to 140 characters. These status messages are called "tweets" or more generally "microposts" and the subscribers are called "followers". Users can also reply or "retweet" messages. Retweets can be comparable to forwarded emails indicated by "RT" character. Moreover, a message can contain a sort of tag preceded by the hash character which is known as a "hashtag" e.g. #tag. Finally, users have the option to follow other users, thus becomes a "followee" of them.

2.1.3 Exploratory User Study

The initial motivation behind this thesis lies in an exploratory user-centered study conducted by Fialho et al. [54]. The goal is to understand the event-related activities (e.g. searching, attending, sharing) and to collect insights about existing Web-based technologies. This study consisted of a user survey completed by 28 participants and two focus-group sessions (10 and 25 participants). The questions were elaborated to assess the perceived benefits and drawbacks of using: event directories, media directories, social networks, and a merger of these services.

Finding and attending an event

Participants reported to discover events mainly through invitations, recommendations, friends' posts or some traditional media (e.g. news articles, ads, etc). They

[6] http://www.flickr.com

also refer to previously attended events or venues to find new events, and they use search engines particularly when they knew what to look for. Moreover, it is found that decision about attending an event seems to prioritize some significant constraints such as time, location and price. Social information about which friends will attend an event has also an important role in decision making. Other additional details appear to have slight influence like the case of subjective factors (type, topic, performer). To share their experiences, participants tend to use media directories and social networks by posting comments, photos and few videos.

Use of social directories

According to participants, event directory is the best source to provide a general overview of an event context within a single channel. It also enables a user-friendly event exploration from various views (what, when, where) along with other features (e.g. tickets, comments). However, it appears that the information perceived are often incomplete and insufficient for decision support (lack of media and geographic map). To overcome this issue, media directories have been considered as one valuable outlet that better conveys the event environment based on visual information. Social network sites was also said to enhance event directories by adding communication and sharing features (e.g. comments, invitations) along with other useful details (e.g. attendance, popularity). Besides, some other functionalities have been mentioned to be desirable for reducing the information overload. One functionality supports the event recommendation based on friend's attendance and user interests. Another functionality is to better visualize events by improving search features (e.g. geographic map) and enriching descriptions (e.g. price, attendance).

Recapitulation

TO sum up, lack of coverage of event directories and frustration of being locked in a particular site are the recurrent issues perceived during the study. Participants recognized that there is 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 photos and videos to provide powerful means of identifying several event characteristics. Media is thus useful 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 and recommendation options. Motivated by this study, we decided to build a platform based on Semantic Web technologies in order to integrate information spread in many silos and thus enhance the event coverage.

2.2 Events in Research

In the last few decades, a growing corpus of research has been centered on the notion of "*event*". Such particular attention sheds light on the inherent complex nature of events. This is behind the fact that even the definition of what an event is fails to reach a real consensus. Recently, the growth of social networks along with the technological improvements that made connected devices easy to use, made the user-contributed Web a primary source of information about any kind of real world happening. Studying events on the Web has been the subject of an attractive diversity of research works. Summarizing the various challenges surveyed in these studies, we discern three major key aspects that will drive our strategy to design relevant program or system.

An event is an entity that handles in essence contextual dimensions, each of which is related to one attribute such as time, location, topic and participants. This multi-faceted aspect has driven the design of many programs which aim, for example, to detect events from social media [4, 178] or to explore meaningful relationships between them [35]. Recently, a research study proposed by Ramesh Jain [82] explored the multi-dimensionality to introduce a coherent definition of the so-called "*Web of events*". Indeed, this term has been conceived as the Web in which nodes represent events having informational and experiential attributes with links describing its structure and relationships. Informational attributes provide descriptive metadata of events including title, location, participants, and so on. Experiential attributes describe the sensory data highlighting the event experience such as image and video. Various links can exist in the Web of events such as the one which connects events with experiential attributes. Other links may capture the natural relationships that exist among events such as identical, temporal and causal. Among all the dimensions, it appears that the temporal one has received a substantial attention in research. Several studies in TDT field have been based on time series analysis of media content to identify events. A typical example is the work of Weng et al. [178] that considers an event as a burst of words in a specific time window. Another earlier work proposed by Allen et al. [5] in AI field provides a logic model to represent the temporal relationships between pairs of events.

Also related to the temporal dimension, the second key aspect is the short event lifetime. Broadly speaking, an event is an ephemeral item that only exists between two time instants. This period seems also to be correlated with peaks of users' activities in social networks where people engage in discussions about this event. Such transiency has constrained the design of many real-time systems which should support high scalability and online processing of streaming data. For example, Sakaki et al. [155] proposed a real-time system to identify earthquake events in Twitter. Becker et al. [10] used an online clustering technique to detect in real-time groups of topically similar tweets that correspond to events. Another system perceived to

suffer from the fleeting nature of target items are those which provide personalized recommendations. Unlike classic item (movie, book) recommendation, the system can only acquire a limited history about event participation which induces highly sparse rating data. This is a well-known problem in a recommender system appearing when an item has not received enough ratings to be meaningfully used. Such items require an advanced recommender system such as the one proposed by Cornelis et al. [36] based on the hybridization of existing and popular recommendation techniques (e.g. collaborative filtering, content-based).

The third key aspect is the social information that an event holds. In reality, people regularly attend various events or share their experiences, thus forming a dynamic space of rich social interactions. As such, social networks can be directly constructed from event-centric user activities which can be offline in the physical world or online on the Web. These so called *event-based social networks* have been studies in some research works. For example, Liu et al. [117] proposed a formal definition of an event-based social network, and they extensively studied its underlying properties along with community detection and information diffusion. Liao et al. [116] used them to reveal the latent social relations among users and the implicit users' preferences which are exploited in event recommendation.

2.3 Semantic Web

The current Web, as introduced by Tim Berners-Lee in 1989, is a huge information space mostly represented in interlinked HTML documents. While the interpretation of this information is delegated to human beings, computers serve merely as storage and communication platform. This fact prevents machines from achieving many tasks based on automated data processing such as search and query answering.

As has has been designed for human consumption, the Web still needs a high human involvement to interpret, combine and categorize data. To overcome this limitation, many efforts have been spent in some fields such as Information Retrieval, Machine Learning, and Natural Language Processing (NLP). They have produced complex systems trying to automatically extract meaning from unstructured data. Typical examples are the search engines such as Yahoo[7] and Google[8]. They mainly rely on NLP routines to index data without any knowledge about the meaning of the terms and the relationship between them. Although the emergence of these search engines was a success for the Web, there is still a semantic gap between what the machine understands and what the user knows about the data [128]. This is where Semantic Web intervenes trying to fill the knowledge gap. In this context, Berners-Lee et al. [13] provide the following definition:

The Semantic Web is not a separate Web but an extension of the current

[7] http://www.yahoo.com
[8] http://www.google.com

one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

How to expand a Web of documents for users with a Web of information for machines is the vision of the Semantic Web. The objective is to automate the human data processing without using full-fledged NLP or reasoning methods by giving meaning to resources and linking them. In the Semantic Web, an intelligent document has awareness about its own content, making it exploitable by automatic process. This way will enable machines to answer complex queries which are currently not possible without human involvement. For example, one user may want to find an event that will take place next weekend in Nice covering one specific topic and with suitable price. For this, he/she currently needs to trawl through various websites and look at different fields (e.g. location, topic, price). On the contrary, the answer in the Semantic Web can be provided by an intelligent Web agent that decodes the query and exploits linked data to deliver relevant information.

To realize this vision, a series of technologies and standards have been proposed. They provide ability to add meaning to the Web content and to represent it in a machine understandable format. In the following, we describe some of these standards along wit the trend of Linked Data.

2.3.1 Resource Description Framework (RDF)

Resource Description Framework (RDF) [106] is a recommendation of the World Wide Web Consortium (W3C) that describes the Web resources. In the Semantic Web, a resource is anything that has an identity and it can be a person, document, image, location, etc. Each resource is assigned a Universal Resource Identifier (URI) [12] which is a formatted string to identify an abstract or physical resource. A common type of URI is the Universal Resource Locator (URL) used to identify resources located on the Web.

RDF is originally designed as a simple metamodel for describing information in a direct graph with labeled nodes and arcs. In this model, the nodes represent the Web resources and the arcs represent the properties which link together these resources. Note that a property is a specific aspect, characteristic, attribute, or relation used to describe a resource [106]. In RDF, resources can be described and linked by a set of statements forming a graph, also known as a semantic network. Each statement is a triple which is usually denoted as $< s, p, o >$ and composed of:

- Subject: the resource which the statement refers to. It is identified by a URI.
- Predicate: describes a property of the subject and expresses the relationship between the subject and the object.
- Object: specifies the value of the property. It can be a resource identified by a URI or an atomic value named literal. Note that a literal can be plain or

typed. A plain literal is a string combined with an optional language tag (e.g. "thesis"@en). A typed literal is a string associated with a datatype URI (e.g. "0.52"^^datatypeURI). The datatype URI specifies the datatype of the literal which can be integer, float or date, as defined by the XML Schema Datatype specification[9].

Figure 2.5 depicts an example of RDF graph-based representation about "France" which is identified by a URI on the Web. Note that this URI identifies a subject resource which is assigned the type Country and has *France* as label.

Figure 2.5: Example of RDF representation about France

Several methods exist for serializing the RDF data model. The most common format is RDF/XML. There exist other text-based formats introduced by W3C such as Turtle[10] and N-Triples[11] which are easier to read than RDF/XML. To query the RDF graph, W3C has defined a query language called SPARQL[12]. It contains triple patterns along with their conjunctions (e.g. logical "and") and disjunctions (e.g. logical "or"). It also supports extensible value testing and constraining queries by named RDF graph.

2.3.2 RDF Schema

RDF is a very simple and flexible data model that allows users to describe resources using properties and values. However, it does not provide means to define vocabularies and to specify domain specific classes and properties. Hence, other terms are needed to describe the classes of resources and the relationship between them. As a solution, an extension of RDF called RDF Schema [25] provides a basic vocabulary to interpret RDF statements. RDFS vocabulary simply describes taxonomies of classes and properties and defines very basic restrictions.

In RDF Schema, URIs have as a namespace `http://www.w3.org/2000/01/rdf-schema#` conventionally associated with the prefix *rdfs:*. In summary, (1) A resource

[9] http://www.w3.org/TR/xmlschema-2
[10] http://www.w3.org/TeamSubmission/turtle
[11] http://www.w3.org/TR/n-triples
[12] http://www.w3.org/TR/rdf-sparql-query

is an instance of one class (*rdfs:class*) or more classes where classes are organized in a hierarchy using *rdfs:subClassOf* property; (2) Properties have as class *rdf:Property* and are organized in a hierarchy using *rdfs:subPropertyOf*. Some restrictions on properties are specified such as *rdfs:domain* to define the class of the subject, and *rdfs:range* to define the class of the object.

2.3.3 Ontology Vocabulary

RDF and RDF Schema both have limited expressivity. While RDF describes a simple way to represent structured data, RDF Schema only provides basic hierarchies associated with simple restrictions. However, there is a need for more expressivity to be able to define a formal explicit description of concepts in some complex domains. Therefore, the concept of ontology has been adopted as an extension of RDF Schema with more expressive constructs. Ontology was originally defined by Artificial Intelligence (AI) community as explicit formal specification of a conceptualization in domain of interest [63]. It typically describes the concepts of the domain and the semantic interconnections that hold between them, along with some logic and inference rules. In general, ontology is the reflection of a shared and common understanding of a domain that can be communicated between people and/or machines. For example, given the different websites containing event information, the use of common ontology will enable Web agents to aggregate data and to answer more complex user queries. In the following, we list some core elements of an ontology:

- Class: defines a concept, type or collection in a specific domain. It groups objects that share some properties and are organized in a hierarchy. For instance, in a university domain, the class Student is more specialized than the class Person.

- Individual: also known as instance or object and is a member of a class. For instance, *Nelson Mandela* is an instance of the class Person.

- Property: is a binary relation to describe how classes and individuals can be related to each another. There is datatype property which connects instances with RDF literals, and object property which connects instances of two classes. For example, *hasFather* is an object property that can relate two instances of the class Person.

To model ontologies, the Web Ontology Language (OWL) [62] is the current markup language endorsed by W3C. Compared with RDF and RDFS, OWL defines a vocabulary with additional formal semantics. It provides more relations between classes (e.g. *disjointWith*), logical properties (e.g. *intersectionOf*, *sameAs*) and enumerations (e.g. *oneOf*, *allValuesFrom*), among others.

2.3.4 Linked Open Data

The Semantic Web is predicated on the availability of large amount of structured RDF data, not in isolated islands but as a Web of interlinked data. A major milestone to realize this vision is the Linked Open Data (LOD or Linked Data) project [39] that connects RDF datasets on a large scale. LOD captures a growing knowledge from various domains forming an open "Web of Data" freely available to access, download, and use. Today's LOD comprises billions of RDF triples counting millions of links between data sources. Formally, Linked Data has been defined as about "data published on the Web in such a way that it is machine readable, its meaning is explicitly defined, it is linked to other external datasets, and can in turn be linked to from external datasets" [18].

Linked Data follows the principles outlined by Tim Berners-Lee to publish information on the Web, which are:

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- Include links to other URIs. so that they can discover more things.

Overall, these principles stress on the accessibility and the linkage of data that adhere to the architecture and standards of the Web. Figure 2.6 shows the significant number of published datasets in 2011, covering information from diverse areas such as encyclopedic, government, geographic, entertainment, publications and so on. For instance, DBpedia[13] is one of the largest RDF repository in the Linked Data focusing on extracting multilingual knowledge from Wikipedia. At the time of writing, the English edition of DBpedia consists of 470 millions RDF triples that describe 4.0 million things covering a wide range of topics, and contains 45 million RDF links to several hundred external datasets.

Client applications can access and use RDF links to navigate between datasets and to discover additional information. In order to be part of Linked Data, datasets need to create links to related instances in other datasets. To cope with the large amount of instances, it is a common practice to draw on automated or semi-automated tools or methods to generate links between data sources. Yet, this is still a challenging task and significant research efforts have been devoted to address it.

[13] http://dbpedia.org

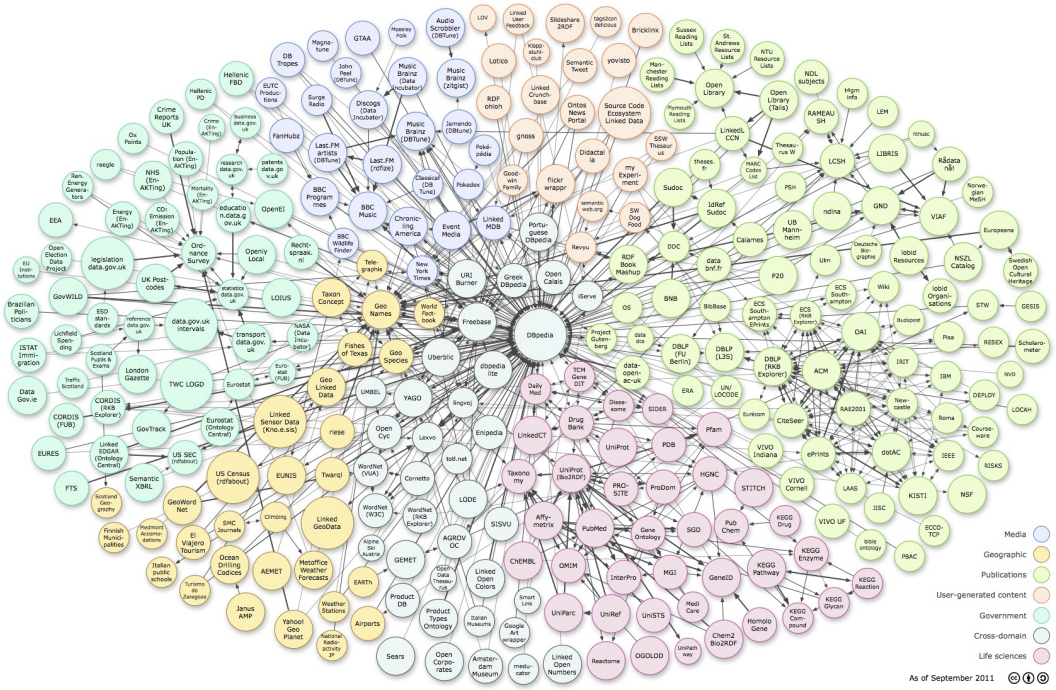

Figure 2.6: Linked Open Data (LOD) Cloud in September 2011

2.4 Evaluation Metrics

In this section, we overview the mostly used evaluation functions in this thesis namely, Precision, Recall and F-score. These measures are widely exploited in data reconciliation field. For a reconciliation task, results can be classified into 4 categories which are: *true positives (tp)*, *true negatives (tn)*, *false positives (fp)* and *false negatives (fn)*. The terms *positive* and *negative* refer to the system's prediction, and the terms *true* and *false* refer to whether this prediction is correctly corresponding to the ground truth. Precision computes the percentage of correctly matched reference pairs (tp) over all matched reference pairs (tp and fp) (Equation 2.1). Recall computes the percentage of correctly matched reference pairs (tp) over pairs of references in the ground truth (tp and fn) (Equation 2.2).

$$Precision = \frac{tp}{tp + fp} \tag{2.1}$$

$$Recall = \frac{tp}{tp + fn} \tag{2.2}$$

In practice, F-score is also popularly used and it combines both precision and recall:

$$F\text{-}score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

2.5 Conclusion

In this chapter, we first reviewed several definitions given to the concept of event and we adopted an interoperable definition that describes essential aspects. Then, some popular social websites hosting event related data have been described. The drawbacks perceived by people to use these websites particularly motivated us to carry out this work. Finally, we detailed the fundamentals of the Semantic Web and the evaluation criterion which will be involved in this thesis.

Part I

Structuring and Linking Event-centric Data on the Web

Overview of Part I

In Part I, we focus on the development of a framework that retrieves and links event-centric information derived from event directories, media platforms and social networks. We capitalize on Semantic Web technologies to ensure a flexible and large-scale integration of disparate data sources, some of which overlap in their coverage. The goal is to provide a support for exploring and selecting events associated with media, and for discovering meaningful connections between them.

In Chapter 4, we present the different steps in building a large dataset called Event-Media which is composed of event descriptions associated with media. These steps include data aggregation and structuring into a unified knowledge model using ontologies. One fundamental requirement is to set a flexible architecture, so that it can easily support the addition of event and media Web services.

In Chapter **??**, we focus on the fourth element of the Linked Data principles which is to link data together. The goal is to explore the implicit overlap of the disparate data sources trying to overcome some well-known types of data heterogeneity. We mainly investigate the following questions: what heuristics are suitable to reconcile event-centric information in Linked Data? How to align structured events with unstructured media content?

Chapter 3

Data Aggregation and Modeling

Along with the advent of Web 2.0, a substantial amount of high-demand information continue to be created and expanded over multiple social services. In particular, information about events, illustrative media and participants' connections are in constant growth. However, this information is often incomplete and locked into the sites, providing limited event coverage and no interoperability of the description. Integrating these distributed data sources into one unified platform is a key factor to enable rich presentation and search of all event content. One major concern that rises in this context is how to ensure a flexible integration that aggregates incrementally different sources of data. The goal is to achieve data integration in reasonable levels of effort and to face the dynamics of social services. In this chapter, we present our framework to integrate data ensuring a certain level of flexibility. Moreover, we explore the intrinsic connection between events and media based on explicit metadata.

3.1 Data Aggregation

In this section, we overview the definition of a Web service. Then, we describe how data from event and media Web services has been collected and interlinked in a flexible way.

3.1.1 Web Service Definition

Web Service has been defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network." [23]. It provides an application-programming interface (API) which describes a specification of remote request-response calls that could be consumed by other systems. In this context, a Web service is sometimes considered as a synonym of Web API. In Web 2.0, the most common API is based on REST architecture in which "the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of *stateless* operations" [23]. REST stands for Representational State Transfer, and it has emerged in the last few years as a predominant Web service design model. It is an alternative way to define Web services and has been introduced in 2000 in the doctoral dissertation of Roy Fielding, one of the principal authors of the HTTP specification. REST strictly refers to a collection of network architecture principles which outline how resources are defined, addressed and transferred over HTTP. With REST, each

resource is referenced with a global identifier (e.g. URI in HTTP). To interact with a resource, an application needs to know the identifier of the resource, the action required and the format of the response. Most of existing Web APIs are currently based on REST architecture, and they define a set of HTTP request methods, along with associated responses usually serialized in XML and JSON formats.

3.1.2 REST-based Scraper

Web services such as Eventful, Last.fm or YouTube become increasingly important for creating Web content mash-ups. Yet, collecting data from these heterogeneous platforms implies the studying of related API specifications which differ in terms of policy, HTTP methods and response schema. To alleviate this task, one typical solution is to design a unified interface that combines various APIs and manages some tasks such as policy management, requests chaining and merging response structures. Some tools providing this solution have been emerged with the aim to save developers' efforts.

For instance, API Blender [60] is an open-source that integrate five platforms, namely: Twitter, Facebook, Flickr, Google+ and YouTube. It describes a Web API using a set of JSON serialized objects including the definition of access policies and API methods. For example, the "Policy" object describes the number of requests per hour and the too-many-calls response code. Although API Blender supports a high flexibility to collect data, it does not address the difference between response schemata. Another tool is the media collector developed for Media Finder application [147]. It enables a parallel key-search over a variety of social networks and exports results in a unified output. It is based on a common alignment response schema in order to be agnostic of a particular social network. The schema describes a set of metadata such as url, type (e.g. photo or video), message (e.g. description of media item), etc. However, there is no support for policy management and the response schema provides very basic information.

In order to ensure a flexible data collection, there is a need for a unified interface that retrieves data from event and media Web services, and exploits the similarity between their Web APIs. We propose the framework[1] illustrated in Figure 3.1 and composed of two main components: the Unified REST Module and the Scraping Processor, described as following.

Unified REST Module

It is based on a RESTful service that allows unifying various Web APIs by exploiting their commonality in terms of HTTP methods, objects and input parameters. Each source API (e.g. Eventful API) is associated with a descriptor file serialized in JSON which provides useful information to handle REST requests. The descriptor file

[1] http://eventmedia.eurecom.fr/scrap

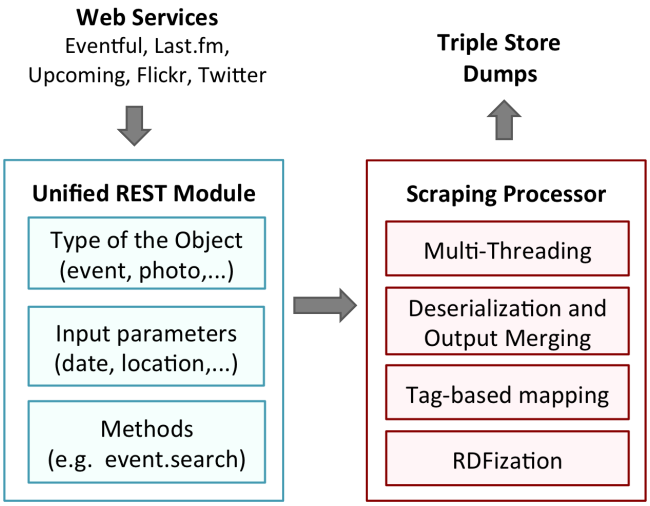

Figure 3.1: Rest-based Scraper Architecture

contains global parameters such as the API key, the API root path and a dictionary of URLs, as depicted in Listing 3.1. In addition, the descriptor file contains an array of query objects. Each query object represents a mapping between our REST URL pattern and the source API URL pattern that describes a REST method and its input parameters. An example of a query object is depicted in Listing 3.2.

Listing 3.1: Global parameters in Last.fm descriptor file

```
{
    "APIName": "Lastfm",
    "APIRootURL": "http://ws.audioscrobbler.com/2.0/?method=",
    "APIKey": "c650...",

    "Prefixes":{
                "publisher": "http://www.last.fm",
                "event": "http://www.last.fm/event/",
                "venue": "http://www.last.fm/venue/",
                "agent": "http://www.last.fm/music/"
            }
}
```

In order to manage the requests chaining, each query object has a type which is used to retrieve the description of main elements (e.g. event, photo, video), and to perform sub-queries that fetch addition information (artist, attendee, etc.). We have defined our own REST methods to search for events, photos and videos, respectively. These methods have as input a set of parameters such as the original sources (e.g. last.fm, eventful, etc.) and other additional filters (e.g. category, location, date, etc.). Thus, the user can request in parallel multiple Web services by specifying the list of sources into one request. This RESTful service is flexible enough, so that new methods can be conveniently created and a new similar REST-inspired Web API can be simply integrated by adding the associated JSON descriptor file.

Listing 3.2: Query object for collecting events in Last.fm

```
"Query": [
            {
              "Type": "events",
              "Method": "{0}geo.getevents&api_key={1}",
              "Inputs": [
                  {
                     "Name": "Location",
                     "Format": "&location={0}",
                     "Required" : "true"
                  },
                  {
                     "Name": "LocationRadius",
                     "Format": "&lat={0}&long={1}&distance={2}",
                     "Required" : "true"
                  },
                  {
                     "Name": "PageNumber",
                     "Format": "&page={0}"
                  },
                  {
                     "Name": "PageSize",
                     "Format": "&limit={0}"
                  }
              ]
            }
  ]
```

Scraping Processor

It is designed to manage requests and process data. It provides a scraping engine to enable multi-threading, where each new request is associated with a thread instance of scraping process. This engine allows only a limited number of thread processes in parallel to respect the Web APIs limits. Moreover, the Scraping Processor handles other tasks for processing data, starting from JSON de-serialization to RDF conversion and loading into a triple store. More precisely, data retrieved is de-serialized and exported into a common schema providing descriptions of a set of objects, namely; event, location, agent, user, photo and video. Then, we employ a tag-based mapping consuming some metadata in order to establish links between events and media (details in Section 3.1.3). This framework is meant to ease the addition of new services for collecting events and media. It also offers other REST methods for monitoring tasks such as tracking or stopping the current scraping processes.

3.1.3 Explicit Linkage

The explicit linkage of resources is straightforward in the presence of shared keys (ISBN, hashtag, etc). Thus, we explore the overlap in metadata that exists between some event and media services.

- Last.fm and Upcoming with Flickr: Explicit relationships between events and photos exist in Flickr using machine tags such as `lastfm:event=ID` where `ID` is the identifier of a specific event (Figure 3.2). These tags are used as filters when searching for photos. Then, each photo resource is linked with the event resource that refers to it.

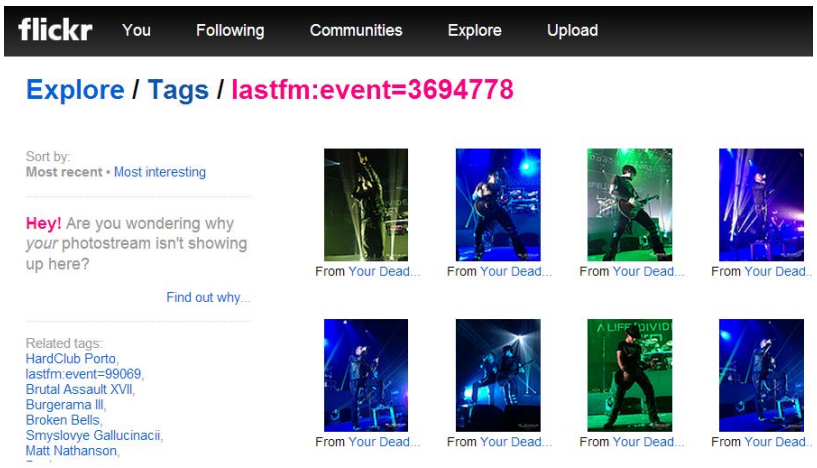

Figure 3.2: Photos with a machine tag identifying one Last.fm event

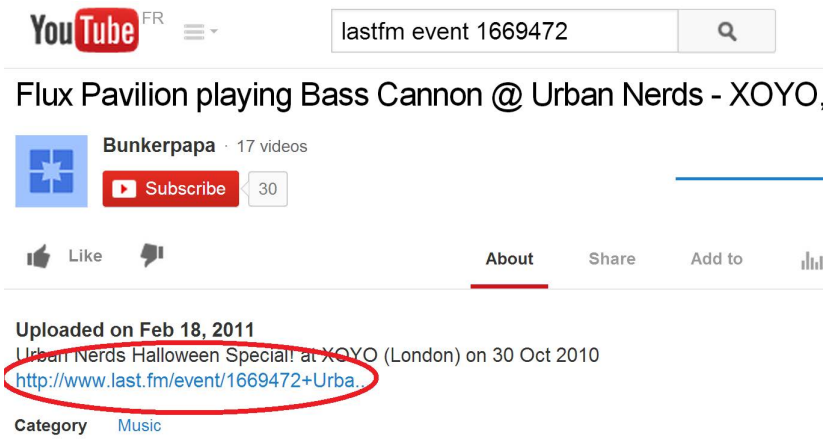

Figure 3.3: Video description including a Last.fm event URL

- Last.fm with YouTube: Similarly, some of YouTube videos have a description including a Last.fm event URL. Thus, videos can be retrieved by a simple keyword search such as "lastfm event". An event identifier could also be added when collecting videos for a specific event. Figure 3.3 illustrates an example of video that will be associated with the event resource identified by *ID=1669472* in Last.fm.

- Lanyrd with Twitter: We also benefit from the overlap between Lanyrd and Twitter, where a hashtag associates each tweet with its related conference. These hashtags are provided by Lanyrd website as depicted in Figure 3.4

Figure 3.4: Lanyrd conference associated with the Twitter hashtag "#uxim2014"

3.1.4 Real-time Scraping

New events are taking place everyday and people keep sharing an ever increasing amount of related media. Such evolution requires a real-time processing that retrieves fresh data and updates the triple store. To achieve this, we developed a live extractor which consumes the feeds provided by some Web services. More specifically, we use the Flickr feeds[2] including the tag "*:event=". Then, a scheduled process reads the feeds every 10 minutes, and trigger accordingly the scraping requests to retrieve the descriptions of events and photos. On an average week, we observe 2000 new photos associated with 160 events (Figure 3.5). Similarly, we also use the Lanyrd feeds[3] that provides fresh conference information including the main hashtag required to retrieve related tweets.

[2] http://api.flickr.com/services/feeds/photos_public.gne?tags=*:event
[3] http://api.lanyrd.com/conferences

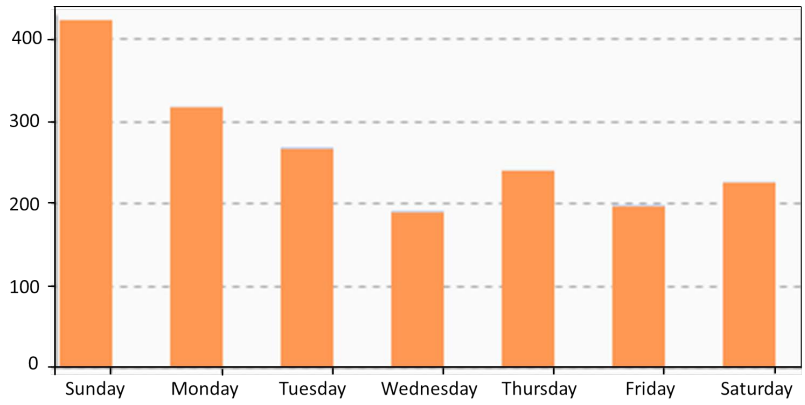

Figure 3.5: # Photos with "*:event=" tag posted in Flickr per days of the week

3.2 Web Dashboard

A Web dashboard has been developed in order to offer graphically and helpful functionalities that help monitor the scraping task. The dashboard is available online at http://eventmedia.eurecom.fr/dashboard. The *Collect* menu provides practical widgets to help build a query by filtering some parameters and provides an option to request the REST-based scraper (Figure 3.6).

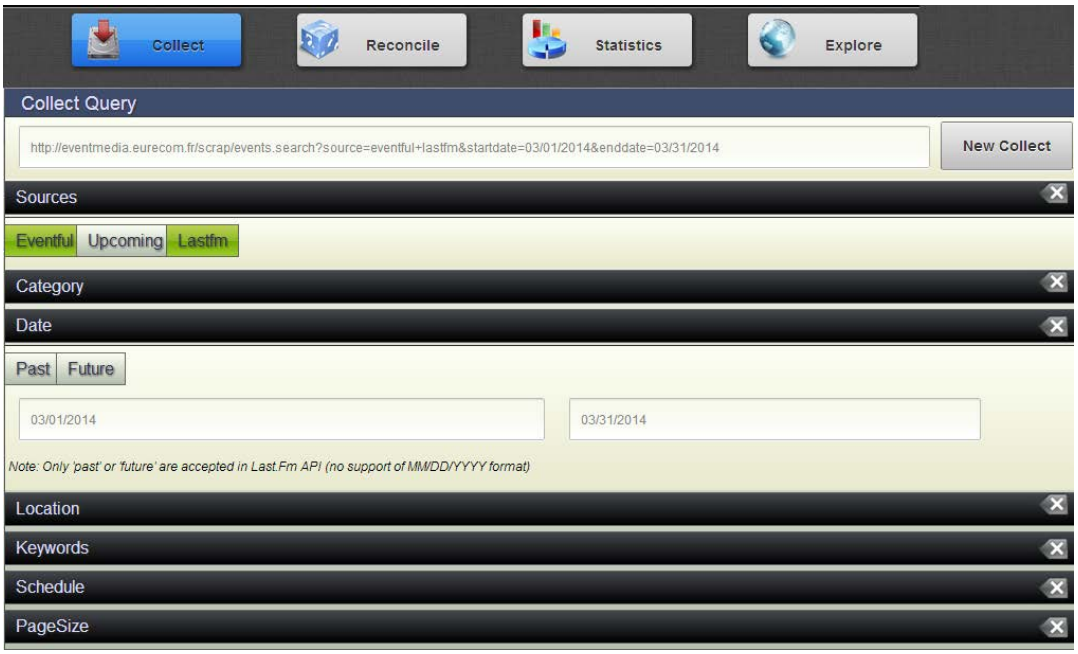

Figure 3.6: Collect Events Mode - Building a query

In order to ensure a permanent progress visualization of the scraping processes, a timer has been set to request the progress service of our framework and to update

Figure 3.7: Collect Media Mode - Monitoring Scraping Processes

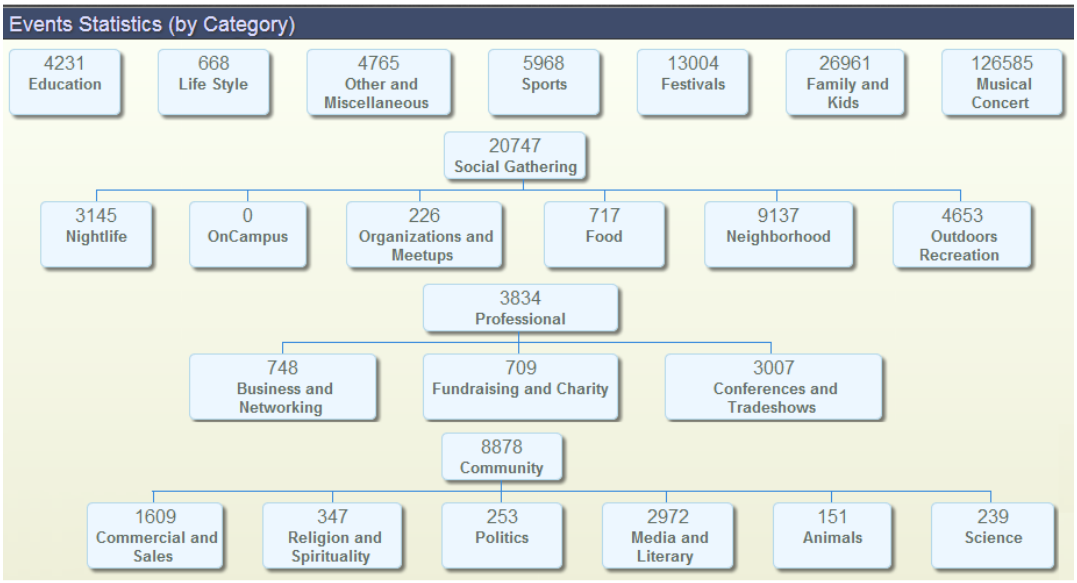

Figure 3.8: Statistics Mode - Number of events per category

accordingly the dashboard (Figuree 3.7). The same timer also updates the console log section by requesting the log service which provides status messages in different types (Debug, Warning, Error). Finally, the dashboard provides *Statistics* view to show useful information about the dataset such as the number of collected instances per type. Figure 3.8 depicts, for example, the number of events per each category. Technically, the languages used are HTML 5 and Javascript with the simple and powerful library jQuery UI. Google maps API and Charts were also used to provide data visualization on maps or to draw chart visualizations.

3.3 Semantic Data Modeling

The motivation behind the use of Semantic Web technologies is their prominent success to provide a flexible support for large-scale data integration. Indeed, a long standing challenge in information systems is to integrate data from distributed, heterogeneous, and autonomous data sources. This refers to the problem of combining data spread across different sources, and providing the user with a unified view of these data. The crucial task lies in forming the relations between the heterogeneous data sources and a global schema representing the unified view. Data sources, some of them being available on the Web, are autonomously designed and operated. As a consequence, they use different systems (e.g. flat files, relational database), data models and access queries. Combining these distributed sources within one application needs an additional layer. This layer has to dynamically integrate data and facilitate interoperability between different structures. In the literature, several integration layers have been proposed as joint efforts of research communities from various fields such as Database, Artificial Intelligence and Semantic Web. One solution widely adopted in recent years is the use of ontologies, which is favored in various disciplines such as biology, medicine and e-government [158]. In the context of Web services, Szomszor et al. [168] have shown the efficiency of ontology-based representation to achieve data harmonization when a mismatch in data formats occurs. The underlying goal of using ontology is to provide a conceptual model that can be shared by different applications. There is an emphasis on knowledge reuse and on the creation of common ontologies which can be extended for more specific applications. This has led to different vocabularies allow describing resources across various domains and facilitate semantic interoperability of metadata. In our case, we use ontologies to enable large-scale integration of data provided by event and media Web services. But, what vocabularies are suitable for representing relationships between events and related entities such as time, location, agent and media?

Given the event definition introduced in Section 2.1.1 and the intrinsic connection between events and media, we consider events as:

- A natural way for referring to any observable occurrence grouping persons, places, times and activities that can be described [160].

- Observable experiences that are often documented by people through different media (e.g. videos, photos and tweets).

In order to formalize this definition, we propose the following ontological models that represent events as well as the related media.

3.3.1 Event Modeling: the LODE Ontology

To represent events, we use the LODE ontology[4] which is a minimal model that encapsulates the most useful event properties [160]. LODE complies with the event definition provided in Section 2.1.1. It is not yet another "event" ontology *per se*. It has been designed as an *interlingua* model that solves an interoperability problem by providing a set of axioms expressing mappings between existing event models. Hence, the ontology contains numerous OWL axioms stating classes and properties equivalence between event models such as EO [144], CIDOC-CRM [49] and ABC [103] to name a few.

Overall, the goal of LODE is to enable an 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. "Factual" relations within and among events are intended to represent intersubjective "consensus reality" and thus are not necessarily associated with a particular perspective or interpretation. We use the LODE ontology together with properties from FOAF [26], Dublin Core [21], and vCard [79]. Our strategy is to separate events from their interpretations with an emphasis on factual aspects, a design approach different from the other event models.

Figure 3.9 depicts the LODE model of the event identified by *ID=3163952* on Last.fm. More precisely, it indicates that an event categorized as a `Concert` has been given on the `21th of May 2012 at 12:45 PM` in `The Paramount Theater`, featuring the `Snow Patrol` rock band and having participant named `earthcapricor`. This event also exists in Upcoming directory but with another identifier *ID=3163952*. To sum up, a graph representing an event contains the category of the event, a text description, a date (instant or interval represented with OWL Time [74]), a location in terms of geographical coordinates (latitude, longitude) and a URI of the venue, and finally the agents (e.g. artists) and attendees involved. A graph representing an agent is composed of a label, description (e.g. biography), tags and often a photo. A graph for location contains a label and different address fields (e.g. street , city, postal code, country). A graph describing a user contains a label, user's real name and often an avatar.

Finally, we propose to organize events in a taxonomy that solves the interoperability of existing classifications. In general, events are categorized in lightweight

[4] http://linkedevents.org/ontology/

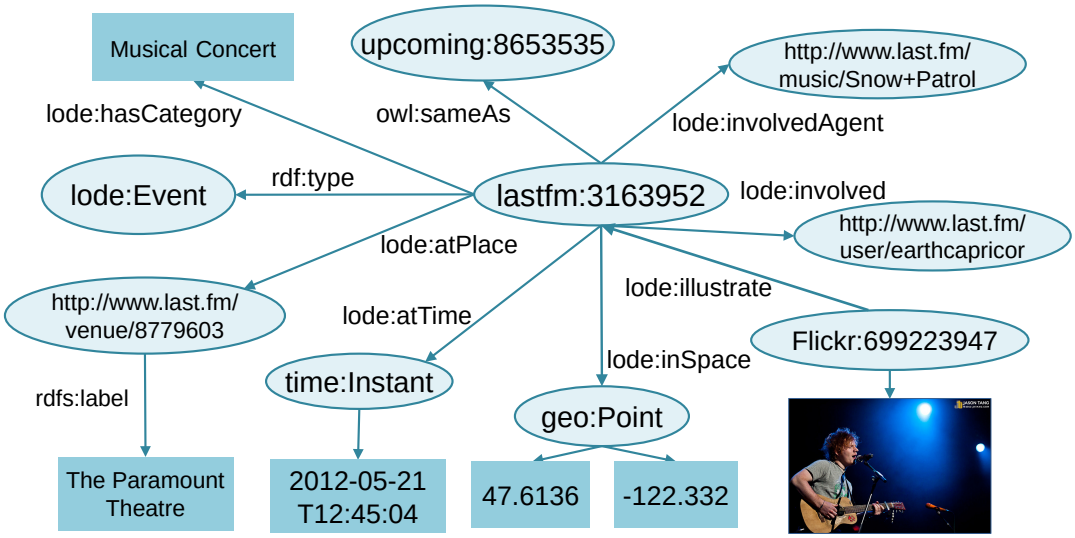

Figure 3.9: The *Snow Patrol Concert* described with LODE ontology

taxonomies that provide facets when browsing event directories. We manually analyzed the event taxonomies used in various websites, namely Facebook, Eventful, Upcoming, LinkedIn[5], Eventbrite[6] and Ticketmaster[7], and we used card sorting techniques in order to build a rich SKOS thesaurus of event categories. SKOS [130] stands for Simple Knowledge Organization System. It provides a vocabulary to represent knowledge organization systems. Such representations include classification schemes, taxonomies and other structured controlled vocabularies. Our SKOS thesaurus contains axioms expressing mapping relationships between the different event taxonomies on the Web. The event taxonomy in our own namespace is accessible online at `http://data.linkedevents.org/category`.

3.3.2 Media Modeling

In order to represent media, we use two popular vocabularies namely, the W3C Media Resource Ontology [109] and the SIOC vocabulary [14]. The link between events and media is realized through the `lode:illustrate` property.

The Media Resource ontology is a W3C initiative that defines a core vocabulary to cover most commonly used annotation properties of media resources (e.g. image, audio, video). Such properties include different types of metadata such as locator, creation date, genre, rating, thumbnails, among others. Media fragments can also be defined to have a smaller granularity and attach keywords or formal annotations to parts of a media item. The ontology contains a formal set of axioms to define the mapping between different metadata formats for multimedia. We use this ontology

[5] `http://www.linkedin.com`
[6] `http://www.eventbrite.com`
[7] `http://www.ticketmaster.com`

together with properties from SIOC, FOAF and Dublin Core to convert into RDF the description of Flickr photo (figure 3.10) and YouTube video. More information can be attached to the URI of the photo or video creator having `sioc:UserAccount` as RDF type.

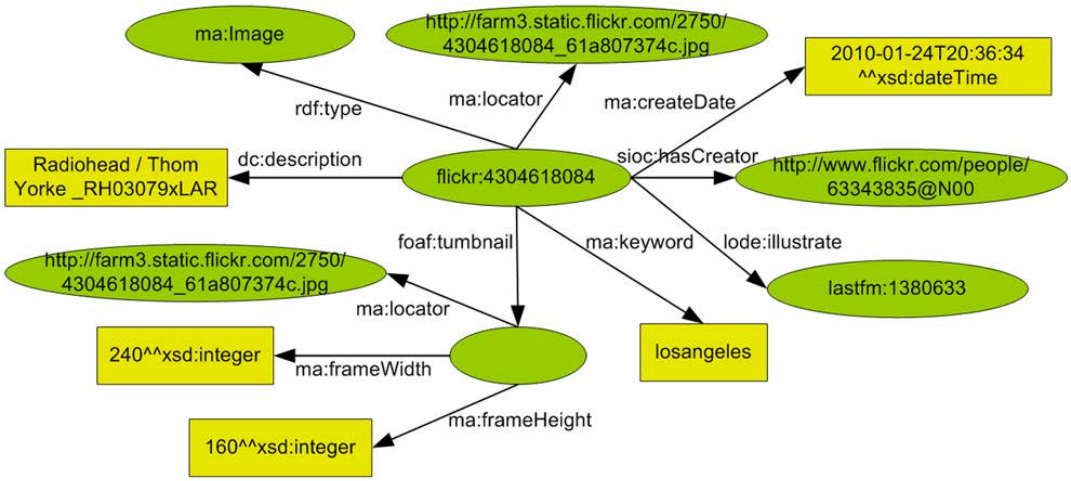

Figure 3.10: A photo taken at the *Radiohead Haiti Relief Concert* described with the Media Ontology

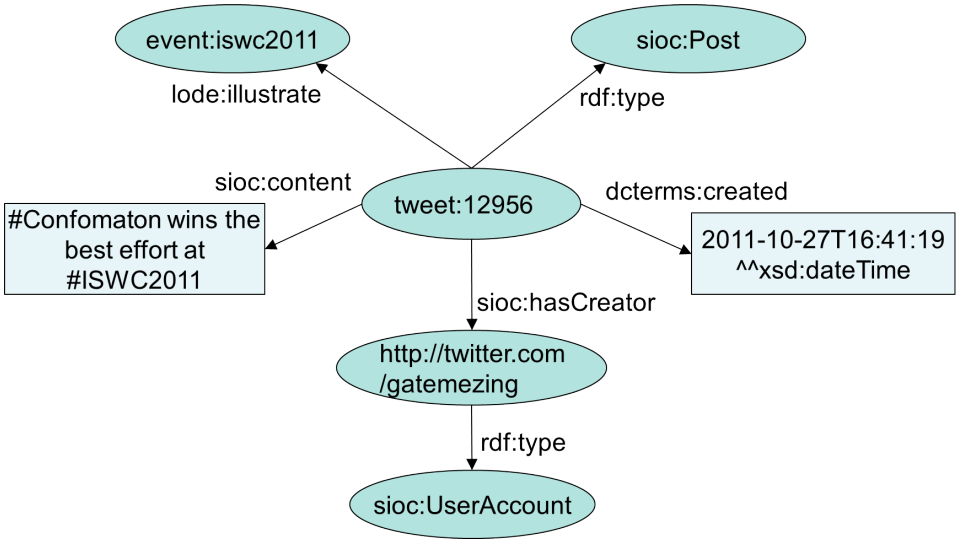

Figure 3.11: RDF modeling of microposts using the SIOC Ontology

SIOC stands for Semantically-Interlinked Online Communities. It provides a core ontology about the main concepts required to describe information about online communities (e.g. wikis, blogs). Such information can include post title, author, keywords, date or the full post text in community sites. SIOC becomes a standard way to model the underlying structure of the user-generated content from social

media sites. We use it together with Dublin Core properties to convert into RDF the description of microposts (Figure 3.11).

3.4 EventMedia Dataset

The so-called EventMedia dataset contains data that we collected from four public event directories (Last.fm, Eventful, Upcoming and Lanyrd), and from three public media directories (Flickr, Youtube, Twitter) [96]. It consists of more than 30 millions RDF triples providing descriptions of events and related media based on LODE, Media Resource and SIOC ontologies. EventMedia is a hub in the Linked Data cloud since September 2010. We mint new URIs into our own namespace such as for events (http://data.linkedevents.org/event/) and agents (http://data.linkedevents.org/agent/). All URIs are dereferencable and served as static RDF files serialized in many formats such as RDF/XML, N3 and N-Triples. They are also accessible using a SPARQL endpoint[8] and a RESTful API[9] powered by the Linked Data API (detailed in Section **??**). Table 3.1 provides an overview about the number of resources per type and source, and Figure 3.12 illustrates the main components of EventMedia.

		Event	**Agent**	**Location**	**Media**	**User**
Event Sites	Last.fm	69,185	81,006	18,653	7,795	213,351
	Upcoming	29,418	78	14,372	29	23,977
	Eventful	84,225	11,226	30,572	15,532	547
	Lanyrd	2,151	-	624	-	-
Media Sites	Flickr	-	-	-	1,879,343	25,219
	Youtube	-	-	-	517	-
	Twitter	-	-	-	1,060,879	267,138

Table 3.1: Number of different resources in EventMedia dataset per type and source

[8] http://eventmedia.eurecom.fr/sparql
[9] http://eventmedia.eurecom.fr/rest/{resourceId}

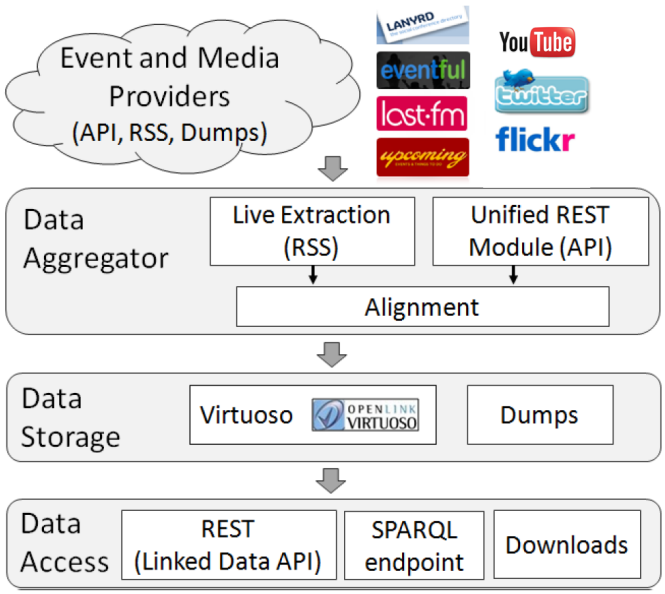

Figure 3.12: Overview of the EventMedia components

3.5 Conclusion

In this chapter, we described our framework designed to aggregate data with the aim to ensure a certain level of flexibility. We also exploited the Semantic Web technologies to integrate at large scale the information contained in event and media directories. As for the semantic modeling, our design is based on the LODE, Media Resource and SIOC ontologies used to describe events and different types of media (e.g. photo, video, micropost). Data collected is converted to RDF and then stored in EventMedia dataset. Ultimately, we aim at providing an event-based environment to deliver enriched views and enhance the event discovery.

Chapter 4

Objective Linked Data Quality

In the last few years the Semantic Web gained a momentum supported by the introduction of many related initiatives like the Linked Open Data (LOD)[1]. From 12 datasets cataloged in 2007, the Linked Open Data cloud has grown to nearly 1000 datasets containing more than 82 billion triples. Data is being published by both public and private sectors and covers a diverse set of domains from life sciences to military. This success lies in the cooperation between data publishers and consumers where users are empowered to find, share and combine information in their applications easily.

We are entering an era where open is the new default. Governments, universities, organizations and even individuals are publicly publishing huge amounts of open data. This openness should be accompanied with a certain level of trust or guarantees about the quality of data. The Linked Open Data is a gold mine for those trying to leverage external data sources in order to produce more informed business decisions [24]. However, the heterogeneous nature of sources reflects directly on the data quality as these sources often contain inconsistent as well as misinterpreted and incomplete information.

Traditional data quality is a thoroughly researched field with several benchmarks and frameworks to grasp its dimensions [86, 165, 175]. Data quality principles typically rely on many subjective indicators that are complex to measure automatically. The quality of data in indeed realized when it is used [85], thus directly relating to the ability of satisfying users' continuous needs.

Web documents that are by nature unstructured and interlinked require different quality metrics and assessment techniques than traditional datasets. For example, the importance and quality of Web documents can be subjectively calculated via algorithms like Page Rank [107]. Ensuring data quality in Linked Open Data is much more complex. It consists of structured information supported by models, ontologies and vocabularies and contains queryable endpoints and links. This makes data quality assurance a challenge. Despite the fact that Linked Open Data quality is a trending and highly demanded topic, very few efforts are currently trying to standardize, track and formalize frameworks to issue scores or certificates that will help data consumers in their integration tasks.

Data quality assessment is the process of evaluating if a piece of data meets the consumers need in a specific use case [17]. The dimensionality of data quality makes it

[1] http://lod-cloud.net

dependent on the task and users requirements. For example, DBpedia [19] is a knowledge base containing data extracted from structured and semi-structured sources. It is used in a variety of applications e.g. annotation systems [126], exploratory search [123] and recommendation engines [132]. However, DBpedia's data is not integrated into critical systems e.g. life critical (medical applications) or safety critical (aviation applications) as its data quality is found to be insufficient. In this paper, we first propose a comprehensive objective framework to evaluate the quality of Linked Data sources. Secondly, we present an extensible quality measurement tool that helps on one hand data owners to rate the quality of their dataset and get some hints on possible improvements, and on the other hand data consumers to choose their data sources from a ranked set. The aim of this paper is to provide researchers and practitioners with a comprehensive understanding of the objective issues surrounding Linked Data quality.

The framework we propose is based on a refinement of the data quality principles described in [7] and surveyed in [57]. Some attributes have been grouped for more detailed quality assessments while we have also extended them by adding for each attribute a set of objective indicators. These indicators are measures that provide users with quality metrics measurable by tools regardless of the use case. For example, when measuring the quality of DBpedia dataset, an objective metric would be the availability of human or machine readable license information rather than the trustworthiness of the publishers.

Furthermore, we surveyed the landscape of Linked Data quality tools to discover that they only cover a subset of the proposed objective quality indicators. As a result, we extend Roomba which is a framework to assess and build dataset profiles with an extensible quality measurement tool and evaluate it by measuring the quality of the LOD cloud group. The results demonstrate that the general quality of LOD cloud needs more attention as most of the datasets suffer from various quality issues.

This paper is structured as follows: Section 4.1 presents the various data quality assessment methodologies. Section 4.2 presents our framework with its objective quality measures and indicators. Section 4.3 presents our tool for evaluating those indicators. Section 4.4 reviews the existing tools and frameworks in the Linked Open Data quality landscape. Section 7.6 presents concluding remarks and identifies future work.

4.1 Data Quality Assessment

In [57], the authors present a comprehensive systematic review of data quality assessment methodologies applied to LOD. They have extracted 26 quality dimensions and a total of 110 objective and subjective quality indicators. However, some of those objective indicators are dependent on the use case thus there is no clear separation on what can be automatically measured. For example, data completeness is gener-

ally a subjective dimension. However, the authors specified that the detection of the degree on which all the real-world objects are represented, detection of number of missing values for specific property and detection of the degree to which instances in the dataset are interlinked are considered as objective indicators given the presence of a gold standard or the original data source to compare with. Moreover, lots of the defined performance dimensions like low latency, high throughput or scalability of a data source were defined as objective but are still dependent on multiple subjective factors like network congestion. In addition, there were some missing objective indicators vital to the quality of LOD e.g. indication of the openness of the dataset.

The ODI certificate[2] provides a description of the published data quality in plain English. It aspires to act as a mark of approval that helps publishers understand how to publish good open data and users how to use it. It gives publishers the ability to provide assurance and support on their data while encouraging further improvements through an ascending scale.

ODI comes as an online and free questionnaire for data publishers focusing on certain characteristics about their data. The questions are classified into the following categories: general information (about dataset, publisher and type of release), legal information (e.g. rights to publish), licensing, privacy (e.g. whether individuals can be identified), practical information (e.g. how to reach the data), quality, reliability, technical information (e.g. format and type of data) and social information (e.g. contacts, communities, etc.). Based on the information provided by the data publisher, a certificate is created with one of four different ratings.

Although ODI is a great initiative, the issued certificates are self-certified. ODI does not verify or review submissions but retains the right to revoke a certificate at any time. The dynamicity of Linked Data makes it also very difficult to update the certificates manually, especially when these changes are frequent and affect multiple categories. There is clearly a need for automatic certification which can be supplemented with some manual input for categories that cannot be processed by machines.

The emerging critical need for large, distributed, heterogeneous, and complex structured datasets identified the necessity to establish industry cooperation between vendors of RDF and Graph database technologies in developing, endorsing, and publishing reliable and insightful benchmark results. The Linked Data Benchmark Council (LDBC)[3] aims to bridge the gap between the industry and the new trending stack of semantic technologies and their vendors. LDBC aims more specifically at developing new benchmarks that will lead to significant progress in scalability, storage, indexing and query optimization techniques to become the de facto standard for publishing performance results. LDBC is a promising initiative, but it is still work in progress with the final report expected on the first quarter of 2015.

[2] https://certificates.theodi.org/
[3] http://ldbc.eu/

In [152], the authors propose a methodology for assessing Linked Data quality. It consists of three main steps: (1) requirement analysis, (2) quality assessment and (3) quality improvement. Considering the multidimensionality of data quality, the methodology requires users to provide the details of a use case or a scenario that describes the intended usage of the data. Moreover, quality issues identification is done with the help of a checklist. The user must have prior knowledge about the details of the data in order to fill this list. Tools implementing the proposed methodology should be able to generate comprehensive quality measures. However, they will require heavy manual intervention and deep knowledge on the data examined. These issues highly affect detecting quality issue on large scale.

Despite all the recent efforts in providing frameworks and tools for data quality in Linked Open Data, there is still no automatic framework for the objective assessment of Linked Data quality.

4.2 Objective Linked Data Quality Classification

The basic idea behind Linked Data is that its usefulness increases when it is more interlinked with other datasets. Tim Berners-Lee defined four main principles for publishing data that can ensure a certain level of uniformity reflecting directly data's usability [170]:

- **Make the data available on the Web**: assign URIs to identify things.
- **Make the data machine readable**: use HTTP URIs so that looking up these names is easy.
- **Use publishing standards**: when the lookup is done provide useful information using standards like RDF.
- **Link your data**: include links to other resources to enable users to discover more things.

Building on these principles, we group the quality attributes into four main categories:

- **Quality of the entities** : quality indicators that focus on the data at the instance level.
- **Quality of the dataset**: quality indicators at the dataset level.
- **Quality of the semantic model**: quality indicators that focus on the semantic models, vocabularies and ontologies.
- **Quality of the linking process**: quality indicators that focus on the inbound and outbound links between datasets.

In [7], the authors identified 24 different Linked Data quality attributes. In this paper, we refine these attributes into a condensed framework of 10 objective measures.

Since these measures are rather abstract, we should rely on quality indicators that reflect data quality [55]. In this paper, we transform the quality indicators presented as a set of questions in [7] into more concrete quality indicator metrics. Independent indicators for entity quality are mainly subjective e.g. the degree to which all the real-world objects are represented, the scope and level of details, etc. However, since entities are governed by the underlying model, we have grouped their indicators with those of the modeling quality. Table 1 lists the refined measures alongside their quality indicators. These attributes are presented in the following sections.

Table 4.1: Objective Linked Data Quality Framework

Quality Attribute	Quality Category	ID	Quality Indicator
Completeness	Dataset Level	1	Existence of supporting structured metadata [77]
		2	Supports multiple serializations [57]
		3	Has different data access points
		4	Uses datasets description vocabularies
		5	Existence of descriptions about its size
		6	Existence of descriptions about its structure (MIME Type, Format)
		7	Existence of descriptions about its organization and categorization
		8	Existence of information about the kind and number of used vocabu
	Links Level	9	Existence of dereferencable links for the dataset [77, 121, 64]
	Model Level	10	Absence of disconnected graph clusters [121]
		11	Absence of omitted top concept [77]
		12	Has complete language coverage [121]
		13	Absence of unidirectional related concepts [77]
		14	Absence of missing labels [121]
		15	Absence of missing equivalent properties [88]
		16	Absence of missing inverse relationships [88]
		17	Absence of missing domain or range values in properties [88]
Availability	Dataset Level	18	Existence of an RDF dump that can be downloaded by users [55][77]
		19	Existence of a queryable endpoint that responds to direct queries
		20	Existence of valid dereferencable URLs (respond to HTTP request)
Licensing	Dataset Level	21	Existence of human and machine readable license information [78]
		22	Existence of de-referenceable links to the full license information [78]
		23	Specifies permissions, copyrights and attributions [57]
Freshness	Dataset Level	24	Existence of timestamps that can keep track of its modifications [56]
Correctness	Dataset Level	25	Includes the correct MIME-type for the content [77]
		26	Includes the correct size for the content
		27	Absence of syntactic errors on the instance level [77]
	Links Level	28	Absence of syntactic errors [167]
		29	Use the HTTP URI scheme (avoid using URNs or DOIs) [121]
	Model Level	30	Contains marked top concepts [121]
		31	Absence of broader concepts for top concepts [121]
		32	Absence of missing or empty labels [2, 121]
		33	Absence of unprintable characters [2, 121] or extra white spaces in la
		34	Absence of incorrect data type for typed literals [77, 2]
		35	Absence of omitted or invalid languages tags [166, 121]

Continued on n

Table 4.1 Objective Linked Data Quality Framework

Quality Attribute	Quality Category	ID	Quality Indicator
		36	Absence of terms without any associative or hierarchical relationships
Comprehensibility	Dataset Level	37	Existence of at least one exemplary RDF file [57]
		38	Existence of at least one exemplary SPARQL query [57]
		39	Existence of general information (title, URL, description) for the datas
		40	Existence of a mailing list, message board or point of contact [55]
	Model Level	41	Absence of misuse of ontology annotations [121, 88]
		42	Existence of annotations for concepts [88]
		43	Existence of documentation for concepts [121, 88]
Provenance	Dataset Level	44	Existence of metadata that describes its authoritative information [56]
		45	Usage of a provenance vocabulary
		46	Usage of a versioning
Coherence	Model Level	47	Absence of misplaced or deprecated classes or properties [77]
		48	Absence of relation and mappings clashes [166]
		49	Absence of blank nodes [78]
		50	Absence of invalid inverse-functional values [77]
		51	Absence of cyclic hierarchical relations [163, 166, 121]
		52	Absence of undefined classes and properties usage [77]
		53	Absence of solely transitive related concepts [121]
		54	Absence of redefinitions of existing vocabularies [77]
		55	Absence of valueless associative relations [121]
Consistency	Model Level	56	Consistent usage of preferred labels per language tag [80, 121]
		57	Consistent usage of naming criteria for concepts [88]
		58	Absence of overlapping labels
		59	Absence of disjoint labels [121]
		60	Absence of atypical use of collections, containers and reification [77]
		61	Absence of wrong equivalent, symmetric or transitive relationships [88]
		62	Absence of membership violations for disjoint classes [77]
Security	Dataset Level	63	Uses login credentials to restrict access [57]
		64	Uses SSL or SSH to provide access to their dataset [57]

4.2.1 Completeness

Data completeness can be judged in the presence of a task where the ideal set of attributes and objects are known. It is generally a subjective measure depending highly on the scenario and use-case in hand. For example, an entity is considered to be complete if it contains all the attributes needed for a given task, has complete language coverage [121] and has documentation properties [129, 121]. Dataset completeness has some objective measures which we include in our framework. A dataset is considered to be complete if it:

- Contains supporting structured metadata [77].
- Provides data in multiple serializations (N3, Turtle, etc.) [57].
- Contains different data access points. These can either be a queryable endpoint (i.e. SPARQL endpoint, REST API, etc.) or a data dump file.

- Uses datasets description vocabularies like DCAT[4] or VOID[5].
- Provides descriptions about its size e.g. `void:statItem`, `void:numberOfTriples` or `void:numberOfDocuments`.
- Existence of descriptions about its format.
- Contains information about its organization and categorization e.g. `dcterms:subject`.
- Contains information about the kind and number of used vocabularies [57].

Links are considered to be complete if the dataset and all its resources have defined links [77, 121, 64]. Models are considered to be complete if they do not contain disconnected graph clusters [121]. Disconnected graphs are the result of incomplete data acquisition or accidental deletion of terms that leads to deprecated terms. In addition to that, models are considered to be complete if they have complete language coverage (each concept labeled in each of the languages that are also used on the other concepts) [121], do not contain omitted top concepts or unidirectional related concepts [77] and if they are not missing labels [121], equivalent properties, inverse relationships, domain or range values in properties [88].

4.2.2 Availability

A dataset is considered to be available if the publisher provides data dumps e.g. RDF dump, that can be downloaded by users [55, 77], its queryable endpoints e.g. SPARQL endpoint, are reachable and respond to direct queries and if all of its inbound and outbound links are dereferencable.

4.2.3 Correctness

A dataset is considered to be correct if it includes the correct MIME-type and size for the content [77] and doesn't contain syntactic errors [77]. Links are considered to be correct if they lack syntactic errors and use the HTTP URI scheme (avoid using URNs or DOIs) [121]. Models are considered to be correct if the top concepts are marked and do not have broader concepts (for example having incoming `hasTopConcept` or outgoing `topConceptOf` relationships) [121]. Moreover, if they don't contain incorrect data type for typed literals [77][2], no omitted or invalid languages tags [166, 121], does not contain "orphan terms" (orphan terms are terms without any associative or hierarchical relationships [119]) and if the labels are not empty, do not contain unprintable characters [2, 121] or extra white spaces [166].

4.2.4 Consistency

Consistency implies lack of contradictions and conflicts. The objective indicators are mainly associated with the modeling quality. A model is considered to be consistent

[4] http://www.w3.org/TR/vocab-dcat/
[5] http://www.w3.org/TR/void/

if it does not contain overlapping labels (two concepts having the same preferred lexical label in a given language when they belong to the same schema) [80, 121], consistent preferred labels per language tag [121, 166], atypical use of collections, containers and reification [77], wrong equivalent, symmetric or transitive relationships [88], consistent naming criteria in the model [121, 88], overlapping labels in a given language for concepts in the same scheme [121] and membership violations for disjoint classes [77, 88].

4.2.5 Freshness

Freshness is a measure for the recency of data. The basic assumption is that old information is more likely to be outdated and unreliable [56]. Dataset freshness can be identified if the dataset contains timestamps that can keep track of its modifications. Data freshness could be considered as a subjective measure. However, our concern is the existence of temporal information allowing dataset consumers to subjectively decide its freshness for their scenario.

4.2.6 Provenance

Provenance can be achieved at the dataset level by including metadata that describes its authoritative information (author, maintainer, creation date, etc.), versioning information and verifying if the dataset uses a provenance vocabulary like PROV [11].

4.2.7 Licensing

Licensing is a quality attribute that is measured on the dataset level. It includes the availability of machine readable license information [78], human readable license information in the documentation of the dataset or its source [78] and the indication of permissions, copyrights and attributions specified by the author [57].

4.2.8 Comprehensibility

Dataset comprehensibility is identified if the publisher provides general information about the dataset (e.g. title, description, URI). In addition, if he indicates at least one exemplary RDF file and SPARQL query and provides an active communication channel (mailing list, message board or e-mail) [55]. A model is considered to be comprehensible if there is no misuse of ontology annotations and that all the concepts are documented and annotated [121, 88].

4.2.9 Coherence

Coherence is the ability to interpret data as expected by the publisher or vocabulary maintainer [77]. The objective coherence measures are mainly associated with the

modeling quality. A model is considered to be coherent when it does not contain undefined classes and properties [77], blank nodes [78], deprecated classes or properties [77], relations and mappings clashes [166], invalid inverse-functional values [77], cyclic hierarchical relations [163, 166, 121], solely transitive related concepts [121], redefinitions of existing vocabularies [77] and valueless associative relations [121].

4.2.10 Security

Security is a quality attribute that is measured on the dataset level. It is identified if the publishers use login credentials, SSL or SSH to provide access to their dataset, or if they only grant access to specific users [57].

4.3 An Extensible Objective Quality Assessment Framework

Roomba is built as a Command Line Interface (CLI) application using Node.js. Instructions on installing and running the framework are available on its public Github repository. Related functions are encapsulated into modules that can be easily plugged in/out the processing pipeline. Figure 4.1 shows the main steps which are the following: (i) Data portal identification; (ii) metadata extraction; (iii) instance and resource extraction; (iv) profile validation (v) profile and report generation.

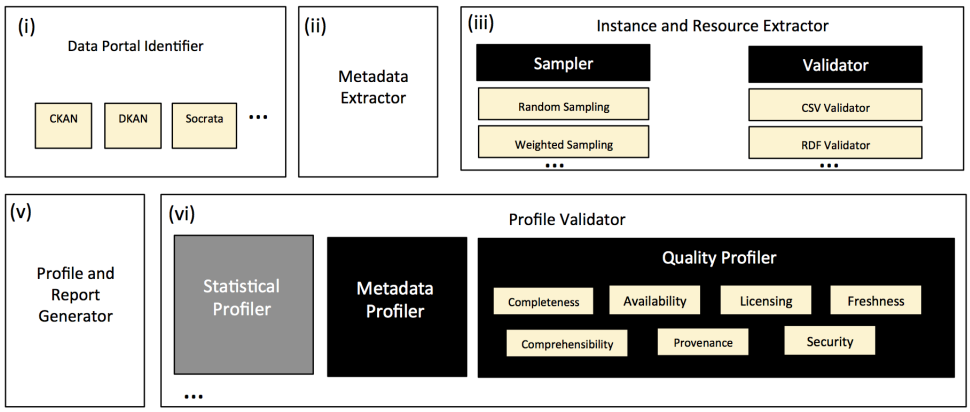

Figure 4.1: Processing pipeline for objective dataset quality assessment

For this paper, we have extended Roomba with a new quality module to measure datasets quality. We have implemented 7 submodules that will check various dataset quality indicators. Various additional quality measures can be easily plugged in/out.

Data portals can be considered as data access points providing tools to facilitate data publishing, sharing, searching and visualization. CKAN[6] is the world's leading open-source data portal platform powering websites and the target of our tool. We

[6] http://ckan.org

have identified that most of the dataset quality issues can be assessed by examining the accompanying dataset metadata. Building a standard metadata model is not the scope of this paper, and since we focus on CKAN-based portals, we assess the quality issues using the CKAN standard model[7]. Table 4.2 shows the various quality indicators checked by our tool.

Quality Indicator	Assessment Method
1	Check if there is a valid metadata file by issuing a `package_show` request to the CKAN API
2	Check if the `format` field for the dataset resources is defined and valid
3	Check the `resource_type` field with the following possible values `file`, `file.upload`, `api`, `visualization`, `code`, `documentation`
4	Check the resources `format` field for `meta/void` value
5	Check the resources `size` or the `triples` extras fields
6	Check the `format` and `mimetype` fields for resources
7	Check if the dataset has a `topic` tag and if it is part of a valid group in CKAN
9	Check if the dataset and all its resources have has a valid URI
18	Check if there is a dereferencable resource with a description containing string *dump*
19	Check if there is a dereferencable resource with `resource_type` of type `api`
20	Check if all the links assigned to the dataset and its resources are dereferencable
21	Check if the dataset contains valid `license_id` and `license_title`
22	Check if the `license_url` is dereferencable
24	Check if the dataset and its resources contain the following metadata fields `metadata_created`, `metadata_modified`, `revision_timestamp`, `cache_last_updated`
25	Check if the `content-type` extracted from the a valid HTTP request is equal to the corresponding `mimetype` field.
26	Check if the `content-length` extracted from the a valid HTTP request is equal to the corresponding `size` field.
28,29	Check that all the links are valid HTTP scheme URIs
37	Check if there is at least one resource with a `format` value corresponding to one of `example/rdf+xml`, `example/turtle`, `example/ntriples`, `example/x-quads`, `example/rdfa`, `example/x-trig`
39	Check if the dataset and its tags and resources contain general metadata `id`, `name`, `type`, `title`, `description`, `URL`, `display_name`, `format`
40	Check if the dataset contain valid `author_email` or `maintainer_email` fields
44	Check if the dataset and its resources contain provenance metadata `maintainer`, `owner_org`, `organization`, `author`, `maintainer_email`, `author_email`
46	Check if the dataset contain and its resources contain versioning information `version`, `revision_id`

Table 4.2: Objective Quality Assessment Methods for CKANbased Data Portals

In our framework, we have presented 30 objective quality indicators related to dataset and links quality. The Roomba quality module is able to assess and score 23 of them. We excluded security related quality indicators as LOD cloud group

[7] http://demo.ckan.org/api/3/action/package_show?id=adur_district_spending

members should not restrict access to their datasets.

4.3.1 Quality Score Calculation

A CKAN portal contains a set of datasets $\mathbf{D} = \{D_1, ...D_n\}$. We denote the set of resources $R_i = \{r_1, ..., r_k\}$, groups $G_i = \{g_1, ..., g_k\}$ and tags $T_i = \{t_1, ..., t_k\}$ for $D_i \in \mathbf{D}(i = 1, ..., n)$ by $\mathbf{R} = \{R_1, ..., R_n\}, \mathbf{G} = \{G_1, ..., G_n\}$ and $\mathbf{T} = \{T_1, ..., t_n\}$ respectively.

Our quality framework contains a set of measures $\mathbf{M} = \{M_1, ..., M_n\}$. We denote the set of quality indicators $Q_i = \{q_1, ..., q_k\}$ for $M_i \in \mathbf{M}(i = 1, ..., n)$ by $\mathbf{Q} = \{Q_1, ..., Q_n\}$. Each quality indicator has a weight, context and a score $Q_i < weight, context, score >$. In Roomba, all the weights are equal and set to 1. However, they can be adjusted manually to rank the quality indicators. Each Q_i of M_i (for i = 1,...n) is applied to one or more of the resources, tags or groups. The indicator context is defined where $\exists Q_i \in \mathbf{R} \cup \mathbf{G} \cup \mathbf{T}$.

The quality indicator score is based on a ratio between the number of violations $\mathbf{V}$ and the total number of instances where the rule applies $\mathbf{T}$ multiplied by the specified weight for that indicator.

$$Q \text{ weightedscore} = (V/T) * Q < weight > \tag{4.1}$$

Q weightedscore is an error ratio. A quality measure score should reflect the alignment of the dataset with respect to the quality indicators. The quality measure score $\mathbf{M}$ is calculated by dividing the weighted quality indicator scores sum by the total number of instances in its context, as the following formula shows:

$$M = 1 - ((\sum_{i=1}^{n} Q \text{ weightedscore})/ \mid Q \text{ context} \mid) \tag{4.2}$$

4.3.2 Experiments and Analysis

In this section, we provide the experiments and evaluation of the proposed framework. All the experiments are reproducible by Roomba and their results are available on its Github repository. We have run the framework on the LOD cloud containing 259 datasets at the time of writing this paper. We ran the instance and resource extractor in order to cache the metadata files for these datasets locally and ran the quality assessment process which took around two hours and a half hour on a 2.6 Ghz Intel Core i7 processor with 16GB of DDR3 memory machine.

```
Dataset Quality Report

completeness quality Score      :   50.22%
availability quality Score      :   26.22%
```

```
licensing quality Score          :   19.59%
freshness quality Score          :   79.49%
correctness quality Score        :   72.06%
comprehensibility quality Score  :   31.62%
provenance quality Score         :   74.07%
Average total quality Score      :   50.47%

=================================================================
                Quality Indicators Average Error %
=================================================================

Quality Indicator : Supports multiple serializations: 11.35%
Quality Indicator : Has different data access points: 19.31%
Quality Indicator : Uses datasets description vocabularies: 88.80%
Quality Indicator : Existence of descriptions about its size: 86.30%
Quality Indicator : Existence of descriptions about its structure: 83.67%
```

Listing 4.1: Excerpt of the LOD cloud group quality report

We found out that licensing, availability and comprehensibility had the worst quality measures scores: 19.59%, 26.22% and 31.62% respectively. On the other hand, the LOD cloud datasets have good quality scores for freshness, correctness and provenance as most of the datasets have an average of 75% for each one of those measures.

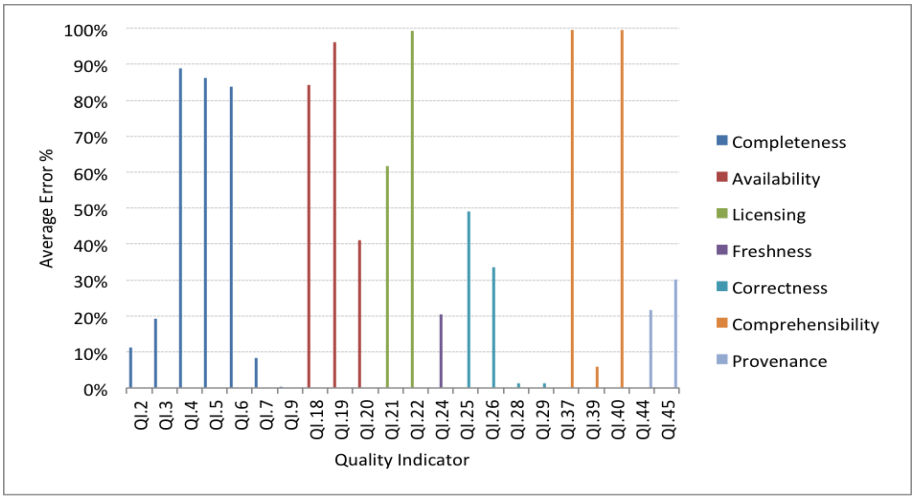

Figure 4.2: Average Error % per quality indicator for LOD group

Figure 4.2 shows the average errors percentage in quality indicators grouped by the corresponding measures. After examining the results, we notice that the worst quality indicators scores are for the comprehensibility measure where 99.61% of the datasets did not have valid exemplary RDF file (QI.37) and did not define valid point of contact (QI.40). Moreover, we noticed that 96.41% of the datasets queryable endpoints (SPARQL endpoints) failed to respond to direct queries (QI.19). After careful examination, we found that the cause was incorrect assignment for metadata

fields. Data publishers specified the resource `format` field as an `api` instead of the specifying the `resource_type` field.

To drill down more on the availability issues, we generated a metadata profile assessment report using Roomba's metadata profiler. We found out that 25% of the datasets access information (being the dataset URL and any URL defined in its groups) has issues related to them (missing or unreachable URLs). Three datasets (1.15%) did not have a URL defined while 45 datasets (17.3%) defined URLs were not accessible at the time writing this paper. Out of the 1068 defined resources 31.27% were not reachable. All these issues resulted in a 26.22% average availability score. This can highly affect the usability of those datasets especially in an enterprise context.

4.4 Linked Data Quality Tools

In this section, we present the results of our survey on the Linked Data quality tools. There exists a number of data quality frameworks and tools that are either standalone or implemented as modules in data integration tools. These approaches can be classified into automatic, semi-automatic, manual or crowdsourced approaches.

4.4.1 Information Quality

RDF is the standard to model information in the Semantic Web. Linked Data publishers can pick from a plethora of tools that can automatically check their RDF files for quality problems[8]. Syntactic RDF checkers are able to detect errors in RDF documents like the W3C RDF Validator[9], RDF:about validator and Converter[10] and The Validating RDF Parser (VRP)[11]. The RDF Triple-Checker[12] is an online tool that helps find typos and common errors in RDF data. Vapour[13] [15] is a validation service to check whether semantic Web data is correctly published according to the current best practices [170].

ProLOD [22], ProLOD++ [1], Aether [135] and LODStats [45] are not purely quality assessment tools. They are Linked Data profiling tools providing clustering and labeling capabilities, schema discovery and statistics about data types and patterns. The statistics are about properties distribution, link-to-literal ratio, number of entities and RDF triples, average properties per entity and average error.

[8] http://www.w3.org/2001/sw/wiki/SWValidators
[9] http://www.w3.org/RDF/Validator/
[10] http://rdfabout.com/demo/validator/
[11] http://139.91.183.30:9090/RDF/VRP/index.html
[12] http://graphite.ecs.soton.ac.uk/checker/
[13] http://validator.linkeddata.org/vapour

4.4.2 Modeling Quality

Reusing existing ontologies is a common practice that Linked Data publishers are always trying to adopt. However, ontologies and vocabularies development is often a long error-prone process especially when many contributors are working consecutively or collaboratively [167]. This can introduce deficiencies such as redundant concepts or conflicting relationships [66]. Getting to choose the right ontology or vocabulary is vital to ensure modeling correctness and consistency.

4.4.2.1 Semi-automatic Approaches

DL-Learner [110] uses supervised machine learning techniques to learn concepts from user-provided examples. CROCUS [32] applies a cluster-based approach for instance-level error detection. It validates identified errors by non-expert users and iterate to reach higher quality ontologies that can be safely used in industrial environments.

4.4.2.2 Automatic Approaches

qSKOS[14] [121] scans SKOS vocabularies to provide reports on vocabulary resources and relations that are problematic. PoolParty checker[15] is an online service based on qSKOS. Skosify [166] supports OWL and RDFS ontologies by converting them into well-structured SKOS vocabularies. It includes automatic correction abilities for quality issues that have been observed by reviewing vocabularies on the Web. The OOPS! pitfall scanner [142] evaluates OWL ontologies against a rules catalog and provides the user with a set of guidelines to solve them. ASKOSI[16] retrieves vocabularies from different sources, stores and displays the usage frequency of the different concepts used by different applications. It promotes reusing existing information systems by providing better management and presentation tools.

Some errors in RDF will only appear after reasoning (incorrect inferences). In [161, 169] the authors perform quality checking on OWL ontologies using integrity constraints involving the Unique Name Assumption (UNA) and the Closed World Assumption (CWA). Pellet[17] provides reasoning services for OWL ontologies. It incorporates a number of heuristics to detect and repair quality issues among disjoint properties, negative property assertions and reflexive, irreflexive, symmetric, and anti-symmetric properties. Eyeball[18] provides quality inspection for RDF models (including OWL). It provides checks for a variety of problems including the usage of unknown predicates, classes, poorly formed namespaces, literal syntax validation, type consistency and other heuristics. RDF:Alerts[19] provides validation for many

[14] https://github.com/cmader/qSKOS
[15] http://www.poolparty.biz/
[16] http://www.w3.org/2001/sw/wiki/ASKOSI
[17] http://clarkparsia.com/pellet
[18] http://jena.sourceforge.net/Eyeball/
[19] http://swse.deri.org/RDFAlerts/

issues highlighted in [77] like misplaced, undefined or deprecated classes or properties.

4.4.3 Dataset Quality

Considering the large amount of available datasets in the Linked Open Data, users have a hard time trying to identify appropriate datasets that suit certain tasks. The most adopted approaches are based on link assessment. Provenance-based approaches and entity-based approaches are also used to compute not only dataset rankings, but also rankings on the entity level.

4.4.3.1 Manual Ranking Approaches

Sieve [127] is a framework for expressing quality assessment and fusion methods. It is implemented as a component of the Linked Data Integration Framework (LDIF)[20]. Sieve leverages the LDIF provenance metadata as quality indicators to produce quality assessment scores. However, despite its nice features, it is only targeted to perform data fusion based on user-configurable conflict resolution tasks. Moreover, since Sieve main input is provenance metadata, it is only limited to domains that can provide such metadata associated with their data.

SWIQA [58] is a framework providing policies or formulas controlling information quality assessment. It is composed of three layers: data acquisition, query and ontology layers. It uses query templates based on the SPARQL Inferencing Notation (SPIN)[21] to express quality requirements. The queries are built to compute weighted and unweighted quality scores. At the end of the assessment, it uses vocabulary elements to annotate important values of properties and classes, assigning inferred quality scores to ontology elements and classifying the identified data quality problems.

4.4.3.2 Crowd-sourcing Approaches

There are several quality issues that can be difficult to spot and fix automatically. In [2] the authors highlight the fact that the RDFification process of some data can be more challenging than others, leading to errors in the Linked Data provisioning process that needs manual intervention. This can be more visible in datasets that have been semi-automatically translated to RDF from their primary source (the best example for this case is DBpedia [19]). The authors introduce a methodology to adjust crowdsourcing input from two types of audience: 1) Linked Data experts, researchers and enthusiasts through a contest to find and classify erroneous RDF triples and 2) Crowdsourcing through the Amazon Mechanical Turk[22].

[20] http://ldif.wbsg.de/
[21] http://spinrdf.org/
[22] https://www.mturk.com/

TripleCheckMate [101] is a crowdsourcing tool used by the authors to run out their assessment supported by a semi-automatic quality verification metrics. The tool allows users to select resources, identify and classify possible issues according to a predefined taxonomy of quality problems. It measures inter-rater agreements, meaning that the resources defined are checked multiple times. These features turn out to be extremely useful to analyze the performance of users and allow better identification of potential quality problems. TripleCheckMate is used to identify accuracy issues in the object extraction (completeness of the extraction value for object values and data types), relevancy of the extracted information, representational consistency and interlinking with other datasets.

4.4.3.3 Semi-automatic Approaches

Luzzu [41] is a generic Linked Data quality assessment framework. It can be easily extended through a declarative interface to integrate domain specific quality measures. The framework consists of three stages closely corresponding to the methodology in [152]. They believe that data quality cannot be tackled in isolation. As a result, they require domain experts to identify quality assessment metrics in a schema layer. Luzzu is ontology driven. The core vocabulary for the schema layer is the Dataset Quality Ontology (daQ) [40]. Any additional quality metrics added to the framework should extend it.

RDFUnit[23] is a tool centered around the definition of data quality integrity constraints [100]. The input is a defined set of test cases (which can be generated manually or automatically) presented in SPARQL query templates. One of the main advantages for this approach is the ability to discover quality problems beyond conventional quality heuristics by encoding domain specific semantics in the test cases.

LiQuate [151] is based on probabilistic models to analyze the quality of data and links. It consists of two main components: A Bayesian Network builder and an ambiguity detector. They rely on data experts to represent probabilistic rules. LiQuate identifies redundancies (redundant label names for a given resource), incompleteness (incomplete links among a given set of resources) and inconsistencies (inconsistent links).

Quality Assessment of Data Sources (Flemming's Data Quality Assessment Tool)[24] calculates data quality scores based on manual user input. The user should assign weights to the predefined quality metrics and answer a series of questions regarding the dataset. These include, for example, the use of obsolete classes and properties by defining the number of described entities that are assigned disjoint classes, the usage of stable URIs and whether the publisher provides a mailing list for the dataset. The main disadvantage for using this tool is the manual intervention which requires deep

[23] http://github.com/AKSW/RDFUnit
[24] http://linkeddata.informatik.hu-berlin.de/LDSrcAss/datenquelle.php

knowledge in the dataset examined. Moreover, the tool lacks support for several quality concerns like completeness or consistency.

LODGRefine [174] is the Open Refine[25] of Linked Data. It does not act as a quality assessment tool, but it is powerful in cleaning and refining raw instance data. LODGRefine can help detect duplicates, empty values, spot inconsistencies, extract Named Entities, discover patterns and more. LODGRefine helps in improving the quality of the dataset by improving the quality of the data at the instance level.

4.4.3.4 Automatic Ranking Approaches

The Project Open Data Dashboard[26] tracks and measures how US government websites implement the Open Data principles to understand the progress and current status of their public data listings. A validator analyzes machine readable files e.g. JSON files for automated metrics like the resolved URLs, HTTP status and content-type. However, deep schema information about the metadata is missing like description, license information or tags.

Similarly on the LOD cloud, the Data Hub LOD Validator[27] gives an overview of Linked Data sources cataloged on the Data Hub. It offers a step-by-step validator guidance to check a dataset completeness level for inclusion in the LOD cloud. The results are divided into four different compliance levels from basic to reviewed and included in the LOD cloud. Although it is an excellent tool to monitor LOD compliance, it still lacks the ability to give detailed insights about the completeness of the metadata and overview on the state of the whole LOD cloud group and is very specific to the LOD cloud group rules and regulations.

Link-based Approaches

The basic idea behind link assessment tools is to provide rankings for datasets based on the cardinality and types of the relationships with other datasets. Traditional link analysis has proven to be an effective way to measure the quality of Web documents search. Algorithms like PageRank [107] and HITS [98] became successful based on the assumption that a certain Web document is considered to have higher importance or rank if it has more incoming links that other Web documents [27][30]. However, the basic assumption that links are equivalent does not suit the heterogeneous nature of links in the Linked Open Data. Thus, the previous approaches fall short to provide reliable rankings as the types of the links can have a direct impact on the ranking computation [171]. The first adaption of PageRank for Semantic Web resources was the Ontology Rank algorithm implemented in the Swoogle search engine [48]. They use a rational random surfing model that takes into account the different types of links between discovered sets and compute rankings based on three levels of

[25] http://openrefine.org/
[26] http://labs.data.gov/dashboard/
[27] http://validator.lod-cloud.net/

granularity: documents, terms and RDF graphs. ReConRank [76] rankings are computed at query time based on two levels of granularity: resources and context graphs. DING [171] adapted the PageRank to rank datasets based on their interconnections. DING can also automatically assign weights to different link types based on the nature of the predicate involved in the link. Broken links are a major threat to Linked Data. They occur when resources are removed, moved or updated. DSNotify[28][69] is a framework that informs data consumers about the various types of events that occur on data sources. Their approach is based on an indexing infrastructure that extracts feature vectors and stores them to an index. A monitoring module detects events on sources and write them to a central event log which pushes notifications to registered applications. LinkQA [64] is a fully automated approach which takes a set of RDF triples as an input and analyzes it to extract topological measures (links quality). However, the authors depend only on five metrics to determine the quality of data (degree, clustering coefficient, centrality, sameAs chains and descriptive richness through sameAs).

Provenance-based Approaches

Provenance-based assessment methods are an important step towards transparency of data quality in the Semantic Web. In [68][29] the authors use a provenance model as an assessment method to evaluate the timeliness of Web data. Their model identifies types of "provenance elements" and the relationships between them. Provenance elements are classified into three types: actors, executions and artifacts. The assessment procedure is divided into three steps: 1) Creating provenance graph based on the defined model 2) Annotating the graph with impact values 3) Calculating the information quality score. In [56] the authors describe a set of provenance-based assessment metrics to support quality assessment and repair in Linked Open Data. They rely on both data and metadata and use indicators like the source reputation, freshness and plausibility. In [67] the authors introduce the notion of naming authority which connects an identifier with the source to establish a connection to its provenance. They construct a naming authority graph that acts as input to derive PageRank scores for the data sources.

Entity-based Approaches

Sindice [43] uses a set of techniques to rank Web data. They use a combination of query dependent and query independent rankings implemented in the Semantic Information Retrieval Engine (SIREn)[30] to produce a final entity rank. Their query dependent approach rates individual entities by aggregating the the score of the matching terms with a term frequency - inverse subject frequency (tf-isf) algorithm. Their query independent ranking is done using hierarchical links analysis algorithms [44]. The combination of these two approaches is used to generate a global weighted

[28] http://www.cibiv.at/~niko/dsnotify/
[29] http://trdf.sourceforge.net
[30] http://siren.sindice.com/

rank based on the dataset, entities and links ranks.

4.4.4 Queryable End-point Quality

The availability of Linked Data is highly dependent on the performance qualities of its queryable end-points. The standard query language for Semantic Web resources is SPARQL. As a result, we focus on tools measuring the quality of SPARQL endpoints. In [28][31] the authors present their findings to measure the discoverability of SPARQL endpoints by analyzing how they are located and the metadata used to describe them. In addition to that, they also analyze endpoints interoperability by identifying features of SPARQL 1.0 and SPARQL 1.1 that are supported. The authors tackled the endpoints efficiency by testing the time taken to answer generic, content-agnostic SPARQL queries over HTTP.

Summary

We notice that there is a plethora of tools (syntactic checkers or statistical profilers) that automatically check the quality of information at the entities level. Moreover, various tools can automatically check the models against the objective quality indicators mentioned. OOPS! covers all of them with additional support for the other common modeling pitfalls in [88]. PoolParty covers also a wide set of those indicators but it targets SKOS vocabularies only. However, we notice a lack in automatic tools to check the dataset quality especially in its completeness, licensing and provenance measures. Table 3 summarizes the automatic dataset quality approaches that have implemented tools (full circle denotes full quality indicator assessment, while half circle denoted partial assessment). As can be seen in this table Roomba covers most of the quality indicators with its focus on completeness, correctness provenance and licensing. Roomba is not able to check the existence of information about the kind and number of used vocabularies (QI.8), license permissions, copyrights and attributes (QI.23), exemplary SPARQL query (QI.38), usage of provenance vocabulary (QI.45) and is not able to check the dataset for syntactic errors (QI.27).

These shortcomings are mainly due to the limitations in the CKAN dataset model. However, syntactic checkers and additional modules to examine vocabularies usage could be easily integrated in Roomba to fix QI.27, QI.8 and QI.45. Roomba's metadata quality profiler can fix QI.23 as we have manually created a mapping file standardizing the set of possible license names and their information[32]. We have also used the open source and knowledge license information[33] to normalize license information and add extra metadata like the domain, maintainer and open data conformance.

[31] http://labs.mondeca.com/sparqlEndpointsStatus/

[32] https://github.com/ahmadassaf/opendata-checker/blob/master/util/licenseMappings.json

[33] https://github.com/okfn/licenses

Tool\Indicator	1	2	3	4	5	6	7	8	9	18	19	20	21	22	23	24	25	26	27	28	29	37	38	39	40	44	45	46	63	64
LOV	●		●	●	◗		◗		●	●		●	◗									●		◗		●		●		
Data.gov	●				◗	●			●			●				◗	●							●		◗				
Roomba	●	●	●	●	●	●	●		●	●	●	●	●	●		●	●	●	●	●		●		●	●	●		●		

Table 4.3: Functional Comparison of Automatic Linked Data quality Tools

4.5 Conclusions and Future Work

In this paper, we have presented a comprehensive objective quality framework applied to the Linked Open Data. We have built upon previous efforts with focus on objective data quality measures. We have identified a total of 64 quality indicators that were mapped when suitable to four main categories (entity, dataset, links, models). We have also surveyed more than 30 different tools that measure different quality aspects of Linked Open Data. We identified several gaps in the current tools and identified the need for a comprehensive evaluation and assessment framework and specifically for measuring quality on the dataset level. As a result, we presented an extension of Roomba (An extensible tool to assess and generate dataset profiles) that covers 82% of the suggested datasets objective quality indicators. Based on our experiments running Roomba on the LOD cloud, we discovered that the general state of the datasets needs attention as most of them have low completeness, provenance, licensing and comprehensibility quality scores.

In future work, we plan to integrate tools assessing models quality in addition to syntactic checkers with Roomba. This will provide a complete coverage of the proposed quality indicators. We also intend to suggest ranked quality indicators to improve the quality report. We also plan to run this tool on various CKAN based data portals and schedule periodic reports to monitor their quality evolution. Finally, at some stage, we plan to extend this tool for other data portal types like DKAN and Socrata.

Conclusion of Part I

In this part, we presented the different steps required to build an event-based environment, harnessing the wealth of information provided by different Web services. We used the Semantic Web technologies for connecting the sparse event and media descriptions, so that they become more discoverable and reusable. Overall, we described how the event-centric data has been extracted, converted, interlinked and published following the Linked Data principles.

First, we developed a framework that offers a simple-to-use and flexible tool to scrap events and related media using some popular Web services. Data is continuously feeding EventMedia, a RDF dataset published in the Linked Data cloud.

Second, we detailed the challenges faced to reconcile data retrieved from heterogeneous sources. Evaluations results show how the event matching is sensitive to the temporal distance, and how an efficient string similarity improves the accuracy. Finally, we tackled the problem of linking microposts with fine-grained events, which represents a tremendous challenge given the extreme noise in media content. An important characteristic has driven the design of our approaches which is the real-time nature of events.

Part II

Exploring the Event Landscape: Applications, Recommendation and Community Detection

Overview of Part II

As the Web of Data contains millions of RDF triples, consuming this knowledge can benefit various tasks such as enrichment, personalization and social analysis.

In Part II, we consume the Linked Data in event domain in order to create Semantic Web applications and to provide valuable solutions for personalization.

In Chapter 5, we present some Semantic Web applications that either support a friendly event browser interface or help create an event with consistent details. We highlight the limitations of existing technologies designed to access and use DF data by common Web developers.

In Chapter 6, we propose a hybrid recommender system built on top of Semantic Web to make personalized suggestions of events. Such system faces a number of challenges due to the inherent complex nature of events.

In Chapter 7, we propose an approach to detect meaningful communities in event-based social network (EBSN). We leverage the event-media links to construct networks based on event-centric users' activities in media platforms.

Chapter 5

Consuming Linked Data in Event Domain

5.1 Introduction

From 12 datasets cataloged in 2007, the Linked Open Data cloud has grown to nearly 1000 datasets containing more than 82 billion triples[1] [?]. Data is being published by both the public and private sectors and covers a diverse set of domains from life sciences to media or government data. The Linked Open Data cloud is potentially a gold mine for organizations and individuals who are trying to leverage external data sources in order to produce more informed business decisions [24]. This success lies in the cooperation between data publishers and consumers. Consumers are empowered to find, share and combine information in their applications easily. However, the heterogeneous nature of data sources reflects directly on the data quality as these sources often contain inconsistent as well as misinterpreted and incomplete metadata information. Considering the significant variation in size, the languages used and the freshness of the data, one realizes that finding useful datasets without prior knowledge is increasingly complicated. This can be clearly noticed in the LOD Cloud where few datasets such as DBPedia [?], Freebase [?] and YAGO [?] are favored over less popular datasets that may include domain specific knowledge more suitable for the tasks at hand. For example, for the task of building context-aware recommender systems in an academic digital library over LOD cloud, popular datasets like Semantic Web Dog Food, DBLP or Yovisto can be favored over lesser known but more specific datasets like VIAF[2] which links authority files of 20 national libraries, list of subject headings for public libraries in Spain[3] or the French dissertation search engine[4].

The main entry point for discovering and identifying datasets is either through public data portals such as DataHub[5] and Europe's Public Data[6] or private search engines such as Quandl[7] and Engima[8]. Private portals harness manually curated data

[1] http://datahub.io/dataset?tags=lod
[2] http://datahub.io/dataset/viaf
[3] http://datahub.io/dataset/lista-encabezamientos-materia
[4] http://datahub.io/dataset/thesesfr
[5] http://datahub.io
[6] http://publicdata.eu
[7] https://quandl.com/
[8] http://enigma.io/

from various sources and expose them to users either freely or through paid plans. The data available is of higher quality but lesser quantity compared to what is available in public portals. Similarly, in some public data portals, administrators manually review datasets information, validate, correct and attach suitable metadata information. This information is mainly in the form of predefined tags such as *media, geography, life sciences* for organization and clustering purposes. However, the diversity of those datasets makes it harder to classify them in a fixed number of predefined tags that can be subjectively assigned without capturing the essence and breadth of the dataset [**?**]. Furthermore, the increasing number of datasets available makes the metadata review and curation process unsustainable even when outsourced to communities.

Data profiling is the process of creating descriptive information and collect statistics about that data. It is a cardinal activity when facing an unfamiliar dataset [**?**]. It helps in assessing the importance of the dataset, in improving users' ability to search and reuse part of the dataset and in detecting irregularities to improve its quality. Data profiling includes typically several tasks:

- **Metadata profiling**: Provides general information on the dataset (dataset description, release and update dates), legal information (license information, openness), practical information (access points, data dumps), etc.
- **Statistical profiling**: Provides statistical information about data types and patterns in the dataset, i.e. properties distribution, number of entities and RDF triples, etc.
- **Topical profiling**: Provides descriptive knowledge on the dataset content and structure. This can be in form of tags and categories used to facilitate search and reuse.

In this work, we address the challenges of automatic validation and generation of descriptive datasets profiles. This paper proposes an extensible framework consisting of a processing pipeline that combines techniques for data portals identification, datasets crawling and a set of pluggable modules combining several profiling tasks. The framework validates the provided dataset metadata against an aggregated standard set of information. Metadata fields are automatically corrected when possible, e.g. adding a missing license URL reference. Moreover, a report describing all the issues highlighting those that cannot be automatically fixed is created to be sent by email to the dataset's maintainer. There exist various statistical and topical profiling tools for both relational and Linked Data. The architecture of the framework allows to easily add them as additional profiling tasks. However, in this paper, we focus on the task of dataset metadata profiling and present our findings by running our framework on the LOD cloud. The results demonstrate that the general state of LOD cloud needs more attention as most of the datasets suffer from bad quality metadata lacking some informative metrics needed to facilitate dataset search. The

noisiest metadata are the access information such as licensing information, resource descriptions as well as resource availability problems.

The remainder of the paper is structured as follows. In Section 6.4, we review relevant related work. In Section 5.3, we describe our proposed framework's architecture and components that validate and generate dataset profiles. In Section 5.4, we present the results when running this tool on the LOD cloud and we summarize the main issues found. Finally, we conclude and outline some future work in Section 7.6.

5.2 Related Work

There exists a considerable amount of tools that tackle specific profiling tasks. For example, [1][135] focus on generating statistical dataset information where in [**?**][**?**] authors use various techniques to attach additional topical information. However, to the best of our knowledge, this is the first effort towards extensible automatic validation and generation of descriptive dataset profiles. For this paper, we will focus on Linked Data metadata profiling tasks. However, one of the advantages of this framework is the ability to easily configure additional profiling tasks e.g. statistical or topical and accommodate different data types e.g. relational.

Data Catalog Vocabulary (DCAT) [**?**] and the Vocabulary of Interlinked Datasets (VoID) [**?**] are concerned with metadata about RDF datasets. There exist several tools aiming at exposing dataset metadata using these vocabularies. In [**?**] authors generate VoID descriptions limited to a subset of properties that can be automatically deduced from resources within the dataset. However, it still provides data consumers with interesting insights. Quality Assessment of Data Sources (Flemming's Data Quality Assessment Tool)[9] provides basic metadata assessment as it calculates data quality scores based on manual user input. The user assigns weights to the predefined quality metrics and answer a series of questions regarding the dataset. These include, for example, the use of obsolete classes and properties by defining the number of described entities that are assigned disjoint classes, the usage of stable URIs and whether the publisher provides a mailing list for the dataset. The ODI certificate[10] on the other hand provides a description of the published data quality in plain English. It aspires to act as a mark of approval that helps publishers understand how to publish good open data and users how to use it. It gives publishers the ability to provide assurance and support on their data while encouraging further improvements through an ascending scale. ODI comes as an online and free questionnaire for data publishers focusing on certain characteristics about their data. Although these approaches try to perform metadata profiling, they are either incomplete or manual.

[9] http://linkeddata.informatik.hu-berlin.de/LDSrcAss/datenquelle.php
[10] https://certificates.theodi.org/

In our framework, we propose a more automatized and complete approach.
The Project Open Data Dashboard[11] tracks and measures how US government websites implement the Open Data principles to understand the progress and current status of their public data listings. A validator analyzes machine readable files e.g. JSON files for automated metrics like the resolved URLs, HTTP status and content-type. However, deep schema information about the metadata is missing like description, license information or tags. Similarly on the LOD cloud, the Data Hub LOD Validator[12] gives an overview of Linked Data sources cataloged on the Data Hub. It offers a step-by-step validator guidance to check a dataset completeness level for inclusion in the LOD cloud. The results are divided into four different compliance levels from basic to reviewed and included in the LOD cloud. Although it is an excellent tool to monitor LOD compliance, it still lacks the ability to give detailed insights about the completeness of the metadata and overview on the state of the whole LOD cloud group and is very specific to the LOD cloud group rules and regulations.
Although the above mentioned tools are able to provide various information about a dataset, there exist no approach that is extensible to combine further information coming from various profiling tools.

Statistical profiling: Calculating statistical information on datasets is vital to applications dealing with query optimization and answering, data cleansing, schema induction and data mining [?] [?] [?]. Semantic sitemaps [?] and RDFStats [?] where one of the first to deal with RDF data statistics and summaries. ExpLOD [?] creates statistics on the interlinking between datasets based on `owl:sameAs` links. In [?] the author introduces a tool that induces the actual schema of the data and gather corresponding statistics accordingly. LODStats [?] is a stream-based approach that calculates more general dataset statistics. ProLOD++ [1] is a Web-based tool that allows LOD analysis via automatically computed hierarchical clustering [?]. Aether [135] generates VoID statistical descriptions of RDF datasets. It also provides a Web interface to view and compare VoID descriptions. LODOP [?] is a MapReduce framework to compute, optimize and benchmark dataset profiles. The main target for this framework is to optimize the runtime costs for Linked Data profiling. In [?] authors calculate certain statistical information for the purpose of observing the dynamic changes in datasets.

Topical Profiling: Topical and categorical information facilitates dataset search and reuse. Topical profiling focuses on content-wise analysis at the instances and ontological levels. GERBIL [?] is a general entity annotation framework that provides machine processable output allowing efficient querying. In addition, there exist several entity annotation tools and frameworks [?] but none of those systems are designed specifically for dataset annotation. In [?], authors created a semantic portal

[11] http://labs.data.gov/dashboard/
[12] http://validator.lod-cloud.net/

to manually annotate and publish metadata about both LOD and non-RDF datasets. In [?], authors automatically assigned Freebase domains to extracted instance labels of some of the LOD Cloud datasets. The goal was to provide automatic domain identification, thus enabling improving datasets clustering and categorization. In [?], authors extracted dataset topics by exploiting the graph structure and ontological information, thus removing the dependency on textual labels. In [?] authors generate VoID and VoL descriptions via a processing pipeline that extracts dataset topic models ranked on graphical models of selected DBpedia categories.

Dataset search can be done without relying on attached metadata (tags and categories). For example, there exist several approaches to create LOD indexes. In [?], authors used VoID descriptions to optimize query processing by determining relevant query-able datasets. In [?], authors created an approximate index structure (QTree) and an algorithm for answering conjunctive queries over Linked Data. SchemEX [?] is a stream-based approach leveraging type and property information of RDF instances to create schema-level indexes.
Semantic search engines like Sindice [44], Swoogle [48] and Watson [?] help in entities lookup but are not designed specifically for dataset search. In [?], authors utilized the sig.ma index [?] to identify appropriate data sources for interlinking. However, the current main source for dataset search and discovery is via data portals. CKAN and DKAN powered data portals rely on attached metadata to provide dataset search features as they run a Solr index on the metadata schemas. Having missing or inconsistent information will affect the search results quality.

5.3 Profiling Data Portals

In this section, we provide an overview of the processing steps for validating and generating dataset profiles. Figure 5.1 shows the main steps which are the following: (i) Data portal identification; (ii) metadata extraction; (iii) instance and resource extraction; (iv) profile validation (v) profile and report generation.

Our framework is built as a Command Line Interface (CLI) application using Node.js. Instructions on installing and running the framework are available on its public Github repository[13]. Related functions are encapsulated into modules that can be easily plugged in/out the processing pipeline. The various steps are explained in details below.

5.3.1 Data Portal Identification

Data portals can be considered as data access points providing tools to facilitate data publishing, sharing, searching and visualization. CKAN[14] is the world's lead-

[13]https://github.com/ahmadassaf/opendata-checker
[14]http://ckan.org

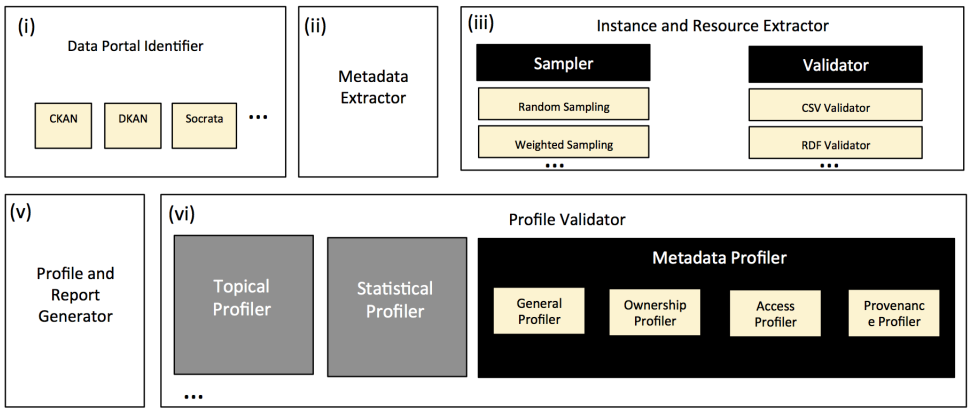

Figure 5.1: Processing pipeline for validating and generating dataset profiles

ing open-source data portal platform powering websites like the DataHub, Europe's Public Data and the U.S Government's open data. Modeled on CKAN, DKAN[15] is a standalone Drupal distribution that is used in various public data portals as well. Socrata[16] helps public sector organizations improve data-driven decision making by providing a set of solutions including an open data portal. In addition to these tradition data portals, there is a set of tools that allow exposing data directly as RESTful APIs like Datatank[17] and Database-to-API[18].

Identifying the software powering data portals is a vital first step to understand the API calls structure. Web scraping is a technique for extracting data from Web pages. We rely on several scraping techniques in the identification process which includes a combination of the following:

- **URL inspection**: Check the existence of certain URL patterns. Various CKAN based portals are hosted on subdomains of the `http://ckan.net`. For example, CKAN Brazil (`http://br.ckan.net`).

- **Meta tags inspection**: The `<meta>` tag provides metadata about the HTML document. They are used to specify page description, keywords, author, etc. Inspecting the `content` attribute can indicate the type of the data portal. We use CSS selectors to check the existence of these meta tags. An example of a query selector is `meta[content*="ckan]` (all meta tags with the attribute content containing the string $CKAN$). This selector can identify CKAN portals whereas the `meta[content*="Drupal"]` can identify DKAN portals.

- **Document Object Model (DOM) inspection**: Similar to the meta tags inspection, we check the existence of certain DOM elements or properties. For

[15]http://drupal.org/project/dkan
[16]http://www.socrata.com
[17]http://thedatatank.com
[18]https://github.com/project-open-data/db-to-api

example, CKAN powered portals will have DOM elements with class names like `ckan-icon` or `ckan-footer-logo`. A CSS selector like `.ckan-icon` will be able to check if a DOM element with the class name `ckan-icon` exists.
The list of elements and properties to inspect is stored in a separate configurable object for each portal. This allows the addition and removal of elements as deemed necessary.

The identification process for each portal can be easily customized by overriding the default function. Moreover, adding or removing steps from the identification process can be easily configured.
After those preliminary checks, we query one of the portal's API endpoints. For example, DataHub is identified as CKAN, so we will query the API endpoint on `http://datahub.io/api/action/package_list`. A successful request will list the names of the site's datasets, whereas a failing request will signal a possible failure of the identification process.

5.3.2 Metadata Extraction

Data portals expose a set of information about each dataset as metadata. The model used varies across portals. However, a standard model should contain information about the dataset's title, description, maintainer email, update and creation date, etc. We divided the metadata information into the following:

General information: General information about the dataset. e.g. title, description, ID, etc. This general information is manually filled by the dataset owner. In addition to that, tags and group information is required for classification and enhancing dataset discoverability. This information can be entered manually or inferred modules plugged into the topical profiler.

Access information: Information about accessing and using the dataset. This includes the dataset URL, license information i.e. license title and URL and information about the dataset's resources. Each resource has as well a set of attached metadata e.g. resource name, URL, format, size, etc.

Ownership information: Information about the ownership of the dataset. e.g. organization details, maintainer details, author, etc. The existence of this information is important to identify the authority on which the generated report and the newly corrected profile will be sent to.

Provenance information: Temporal and historical information on the dataset and its resources. For example, creation and update dates, version information, version, etc. Most of this information can be automatically filled and tracked.

Building a standard metadata model is not the scope of this paper, and since we focus on CKAN-based portals, we validate the extracted metadata against the CKAN standard model[19].

[19] http://demo.ckan.org/api/3/action/package_show?id=adur_district_spending

After identifying the underlying portal software, we perform iterative queries to the API in order to fetch datasets metadata and persist them in a file-based cache system. Depending on the portal software we can issue specific extraction jobs. For example, in CKAN based portals, we are able to crawl and extract the metadata of a specific dataset, all the datasets in a specific group e.g. LOD Cloud or all the datasets in the portal.

5.3.3 Instance and Resource Extraction

From the extracted metadata we are able to identify all the resources associated with that dataset. They can have various types like a SPARQL endpoint, API, file, visualization ,etc. However, before extracting the resource instance(s) we perform the following steps:

- **Resource metadata validation and enrichment**: Check the resource attached metadata values. Similar to the dataset metadata, each resource should include information about its mimetype, name, description, format, valid dereferenceable URL, size, type and provenance. The validation process issue an HTTP request to the resource and automatically fills up various missing information when possible, like the mimetype and size by extracting them from the HTTP response header. However, missing fields like name and description that needs manual input are marked as missing and will appear in the generated summary report.
- **Format validation**: Validate specific resource formats against a linter or a validator. For example, node-csv[20] for CSV files and n3[21] to validate N3 and Turtle RDF serializations.

Considering that certain dataset contains large amounts of resources and the limited computation power of some machines on which the framework might run on, a sampler module is introduced to execute various sample-based strategies detailed in [**?**] where they were found to generate accurate results even with comparably small sample size of 10%.

- **Random Sampling**: Randomly selects resources instances.
- **Weighted Sampling**: Weighs each resources as the ratio of the number of datatype properties used to define a resource over the maximum number of datatype properties over all the datasets resources.
- **Resource Centrality Sampling**: Weighs each resource as the ration of the number of resource types used to describe a particular resource divided by the

[20] https://github.com/wdavidw/node-csv
[21] https://github.com/RubenVerborgh/N3.js

total number of resource types in the dataset. This is specific and important to RDF datasets where important concepts tend to be more structured and linked to other concepts.

However, the sampler is not restricted only to these strategies. Strategies like those introduced in [?] can be configured and applied in the processing pipeline.

5.3.4 Profile Validation

A dataset profile should include descriptive information about the data examined. In our framework, we have identified three main profiling information. However, the extensibility of our framework allows for additional profiling techniques to be plugged in easily i.e. a quality profiling module reflecting the dataset quality. In this paper, we focus on the task of metadata profiling.

Metadata validation process identifies missing information and the ability to automatically correct them. Each set of metadata (general, access, ownership and provenance) is validated and corrected automatically when possible. Each profiler task has a set of metadata fields to check against. The validation process check if each field is defined and if the value assigned is valid.

There exist a bunch of special validation steps for various fields. For example, for ownership information where the maintainer email has to be defined, the validator checks if the email field is an actual valid email address. A similar process is done to URLs whenever found. However, we also issue an HTTP `HEAD` request in order to check if that URL is reachable or not. For the dataset resources, we use the `content-header` information when the request is successfull in order to extract, compare and correct further metadata values like mimetype and content size.

Despite the legal issues surrounding Linked Data licenses [?], it is still considered a gold mine for organizations who are trying to leverage external data sources in order to produce more informed business decisions [24]. In [?] the authors see the potential economic effect unfolding in education, transportation, consumer products, electricity, oil and gas, health care and consumer finance. They estimate the potential annual value enabled by Open Data in these domains to be 3 trillion US Dollars across seven domains. As a result, validating license related information is vital. However, from our experiments, we found out that datasets' license information is noisy. The license names if found are not standardized. For example, Creative Commons CCZero can be also CC0 or CCZero. Moreover,the license URI if found and if de-referenceable can point to different reference knowledge bases e.g. `http://opendefinition.org`. To overcome this issue, we have manually created a mapping file standardizing the set of possible license names and the reference knowledge base[22]. In addition, we

[22]https://github.com/ahmadassaf/opendata-checker/blob/master/util/licenseMappings.json

have also used the open source and knowledge license information[23] to normalize the license information and add extra metadata like the domain, maintainer and open data conformance.

```
{
  "license_id" : ["ODC-PDDL-1.0"],
  "disambiguations" : ["Open Data Commons Public Domain Dedication and License (PDDL)"]
},
{
  "license_id" : ["CC-BY-SA-4.0", "CC-BY-SA-3.0"],
  "disambiguations" : ["cc-by-sa", "CC BY-SA","Creative Commons Attribution Share-Alike"]
}
```

Listing 5.1: License mapping file sample

Statistical profiling

There exist a set of tools designed specifically to provide statistical information about a dataset (see section 2). Providing comprehensive statistical information about a dataset isn't in the scope of this paper. However, to show the extensibility of our framework we provide a simple RDF statistical profiler module that can be easily extended and configured. The information provided for each class is the number: triples, distinct objects, distinct literals, distinct IRI reference objects, distinct blank nodes objects, distinct subjects, distinct IRI reference subjects and distinct blank nodes subjects.

Topical profiling

Similar to the statistical profiler, a detailed survey of the existing tools can be found in the related work section. However, we implement a very basic topical profiler by applying Named Entity Disambiguation (NED) on the textual description and title of a dataset using DBpedia Spotlight [126].

5.3.5 Profile and Report Generation

The validation process highlights the missing information and presents them in a human readable report. The report can be automatically sent to the dataset maintainer email if exists in the metadata.

In addition to the generated report, the enhanced profiles are represented in JSON using the CKAN data model and are publicly available[24].

Data portal administrators need an overall knowledge of the portal datasets and their properties. Our framework has the ability to generate numerous reports of all the datasets by passing formated queries. There are two main set of aggregation tasks that can be run:

[23] https://github.com/okfn/licenses
[24] https://github.com/ahmadassaf/opendata-checker/tree/master/results

- **Aggregating meta-field values**: Passing a string that corresponds to a valid field in the metadata. The field can be flat like `license_title` (aggregates all the license titles used in the portal or in a specific group) or nested like `resource>resource_type` (aggregates all the resources types for all the datasets). Such reports are important to have an overview of the possible values used for each metadata field.

- **Aggregating key:object meta-field values**: Passing two meta-field values separated by a colon `:` e.g. `resources>resource_type:resources>name`. These reports are important as you can aggregate the information needed when also having the set of values associated to it printed.

For example, the meta-field value query `resource>resource_type` run against the LODCloud group will result in an array containing $[file, api, documentation...]$ values. These are all the resource types used to describe all the datasets of the group. However, to be able to know also what are the datasets containing resources corresponding to each type, we issue a key:object meta-field query `resource>resource_type:name`. The result will be a JSON object having the `resource_type` as the key and an array of corresponding datasets titles that has a resource of that type.

```
================================================================================
                        Metadata Report
================================================================================
group information is missing. Check organization information as they can be
    mixed sometimes
organization_image_url field exists but there is no value defined
================================================================================
                        Tag Statistics
================================================================================
There is a total of: 21 [undefined] vocabulary_id fields  100.00%
================================================================================
                        License Report
================================================================================
License information has been normalized !
================================================================================
                        Resource Statistics
================================================================================
There is a total of: 10 [missing] url-type fields  100.00%
There is a total of: 9 [missing] created fields  90.00%
There is a total of: 10 [undefined] cache_last_updated fields  100.00%
There is a total of: 10 [undefined] webstore_last_updated fields  100.00%
There is a total of: 10 [undefined] size fields  100.00%
There is a total of: 10 [undefined] hash fields  100.00%
There is a total of: 10 [undefined] mimetype_inner fields  100.00%
There is a total of: 7 [undefined] mimetype fields  70.00%
There is a total of: 10 [undefined] cache_url fields  100.00%
There is a total of: 6 [undefined] name fields  60.00%
There is a total of: 9 [undefined] webstore_url fields  90.00%
There is a total of: 9 [undefined] last_modified fields  90.00%
There is one [undefined] format field  10.00%
================================================================================
                        Resource Connectivity Issues
================================================================================
```

```
There are 2 connectivity issues with the following URLs:
   - http://dbpedia.org/void/Dataset
================================================================================
                        Un-Reachable URLs Types
================================================================================
There are: 1 unreachable URLs of type [file]
```

Listing 5.2: Excerpt of the DBpedia validation report

5.4 Experiments and Evaluation

In this section, we provide the experiments and evaluation of the proposed framework. All the experiments are reproducible by our tool and their results are available on the its Github repository.
We have run the framework on the LOD cloud containing 259 datasets at the time of writing this paper. We ran the instance and resource extractor in order to cache the metadata files for these datasets locally and ran the validation process which took around one and a half hour on a 2.6 Ghz Intel Core i7 processor with 16GB of DDR3 memory machine.
A CKAN dataset metadata describes three main sections in addition to the core dataset's properties. Those are the groups, tags and resources. Each section contains a set of metadata corresponding to one or more metadata type. For example, a dataset resource will have general information such as the resource name, access information such as the resource url and provenance information such as creation date. The framework generates a report aggregating all the problems in all these sections, fixing field values when possible. Errors can be the result of missing metadata fields, undefined field values or field value errors e.g. unreachable URL or incorrect email address.

Figures 5.2 and 5.3 show the percentage of errors found in metadata fields by section and by information type respectively. We found out that the most erroneous information for the dataset core information were ownership related as 41% were missing or undefined. Datasets resources have the poorest metadata. 64% of the general metadata, all the access information and 80% of the provenance information contained missing or undefined values. Table 5.1 shows the top metadata fields errors in each metadata information type.

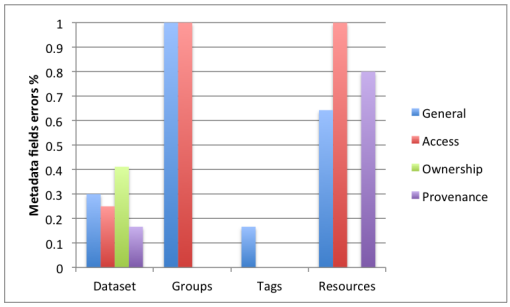

Figure 5.2: Error % by section

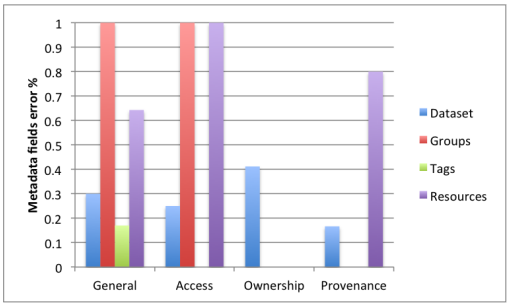

Figure 5.3: Error % by information type

Metadata Field		Error %	Section	Error Type	Auto Fix
General	group	100%	Dataset	Missing	-
	vocabulary_id	100%	Tag	Undefined	-
	url-type	96.82%	Resource	Missing	-
	mimetype_inner	95.88%	Resource	Undefined	Yes
	hash	95.51%	Resource	Undefined	Yes
	size	81.55%	Resource	Undefined	Yes
Access	cahce_url	96.9%	Resource	Undefined	-
	webstore_url	91.29%	Resource	Undefined	-
	license_url	54.44%	Dataset	Missing	Yes
	url	30.89%	Resource	Unreachable	-
	license_title	16.6%	Dataset	Undefined	Yes
Provenance	cache_last_updated	96.91%	Resource	Undefined	Yes
	webstore_last_updated	95.88%	Resource	Undefined	Yes
	created	86.8%	Resource	Missing	Yes
	last_modified	79.87%	Resource	Undefined	Yes
	version	60.23%	Dataset	Undefined	-
Ownership	maintainer_email	55.21%	Dataset	Undefined	-
	maintainer	51.35%	Dataset	Undefined	-
	author_email	15.06%	Dataset	Undefined	-
	organization_image_url	10.81%	Dataset	Undefined	-
	author	2.32%	Dataset	Undefined	-

Table 5.1: Top metadata fields error % by type

We notice that 42.85% of the top metadata problems can be fixed automatically. 44.44% of these problems can be fixed by our tool while the others need tools that are plugged into the data portal. We further present and discuss the results grouped by metadata information type below.

5.4.1 General information

34 datasets (13.13%) did not have valid `notes` values. `tags` information for the datasets were complete except for the `vocabulary_id` as it was missing from all the datasets' metadata. All the datasets `groups` information were missing `display_name, description, title, image_display_url, id, name`. After manual examination, we noticed a clear overlap between group and organization information. Many datasets like `event-media` used the organization field to show group related information (being in LOD Cloud) instead of the publishers details.

5.4.2 Access information

25% of the datasets access information (being the dataset URL and any URL defined in its groups) has issues related to them (missing or unreachable URLs). Three datasets (1.15%) did not have a URL defined (tip, uniprotdatabases, uniprotcitations) while 45 datasets (17.3%) defined URLs were not accessible at the time writing this paper. One dataset did not have resources information (bio2rdfchebi) while the other datasets had a total of 1068 defined resources.
On the datasets resources level, we noticed wrong or inconsistent values in the `size` and `mimetype` fields. 20 (1.87%) resources had incorrect `mimetype` defined, while 52 (4.82%) had incorrect `size` values. These values have been automatically fixed based on the values defined in the HTTP response header. However, 44 datasets have valid `size` field values and 54 have valid `mimetype` field values where they were not reachable, thus providing incorrect information.
15 (68%) fields of all the other access metadata are missing or have undefined values. Looking closely, we noticed that most of these problems can be easily fixed automatically by tools that can be plugged to the data portal. For example, the top six missing fields are the `cache_last_updated`, `cache_url`, `urltype`, `webstore_last_updated`, `mimetype_inner` and `hash` which can be computed and filled automatically. However, the most important missing information which require manual entry are the dataset's `name` and `description` were missing from 817 (76.49%) and 98 (9.17%) resources respectively. A total of 334 resources (31.27%) URLs were not reachable, thus affecting highly the availability of these datasets. CKAN resources can be of various predefined types ($file, file.upload, api, visualization, codeanddocumentation$). The frameowork also breaks down these unreachable resources according to their types. 211 (63.17%) resources did not have valid `resource_type`, 112 (33.53%) were files, 8 (2.39%) and one (0.029%) metadata, example and documentation types.

To have more details about the resources URL types, we created a $key : objectmeta-fieldvalues$ group level report on LOD cloud with `resources>format:title`. This will aggregate the resources format information for each dataset. We found out that only 161 (62.16%) of the datasets valid URLs have SPARQL endpoints defined by

`api/sparql` resource format. 92.27% provided RDF example links and 56.3% provided direct links to RDF down-loadable dumps.

The noisiest part of the access metadata was license information. A total of 43 datasets (16.6%) did not have a defined `license_title` and `license_id` fields, where 141 (54.44%) had missing `license_url` field. However, we managed to normalize 123 (47.49%) of the datasets' license information using the manual mapping file.

5.4.3 Ownership information

Ownership information is divided into direct ownership (author and maintainer) and organization information. Four fields (66.66%) of the direct ownership information were missing or undefined. The breakdown for the missing information is: 55.21% `maintainer_email`, 51.35% `maintainer`, 15.06% `author_email`, 2.32% `author`. Moreover, our framework performs checks to validate existing email values. 11 (0.05%) and 6 (0.05%) of the defined `author_email` and `maintainer_email` fields were not valid email addresses respectively.
For the organization information, two field values (16.6%) were missing or undefined. 1.16% of the `organization_description` and 10.81% of the `organization_image_url` information with two out of these URLs were unreachable.

5.4.4 Provenance information

80% of the resources provenance information were missing or undefined. However, most of the provenance information e.g. `metadata_created, metadata_modified)` can be computed automatically by tools plugged into the data portal. The only field requiring manual entry is the `version` field which was found to be missing from 60.23% of the datasets.

5.5 Conclusion and Future Work

In this paper, we proposed a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. This approach applies several techniques in order to check the validity of the metadata provided and to generate descriptive and statistical information for a particular dataset or for an entire data portal. Based on our experiments running the tool on the LOD cloud, we discovered that the general state of the datasets needs attention as most of them lack informative access information and their resources suffer low availability. These two metrics are of high importance for enterprises looking to integrate and use external linked data.
It has been noticed that the issues surrounding metadata quality affect directly dataset search as data portals rely on such information to power their search index. We noted the need for tools that are able to identify various issues in this metadata

and correct them automatically. We found out that 32.25% of all the metadata information can be automatically fixed, on which 50%of them can be directly fixed by our framework. The rest are mainly provenance information that requires special treatment.
As part of our future work, we plan to introduce workflows that will be able to correct the rest of the metadata either automatically or through intuitive manually-driven interfaces. We also plan to integrate statistical and topical profilers to be able to generate full comprehensive profiles. We also intend to suggest a ranked standard metadata model that will help generate more accurate and scored metadata quality profiles. We also plan to run this tool on various CKAN based data portals, schedule periodic reports to monitor the evolvement of datasets metadata. Finally, at some stage, we plan to extend this tool for other data portal types like DKAN and Socrata.

Chapter 6

Hybrid Event Recommendation

Recommendation in online services has gained momentum during the recent past years as a key factor to deliver personalized content. Reducing the information overload and assisting customers to make decision become part of primary concerns in the e-service area. To this aim, recommender systems attempt to provide efficient filters that decode the user interests, and optimize accordingly the information perceived. To help these systems predict items of interest, various clues are available ranging from a user profile, explicit ratings, to past activities and social interactions. For more details, Appendix D describes two popular recommendation techniques, namely the content-based recommendation and the collaborative filtering.

Integrating a recommender system in event-based services is a key advantage to attract people attending events and to promote face-to-face social interactions. Indeed, the event recommendation can draw on different features such as the user preferences (ratings, likes, etc.), the attended events (visited places, involved artists), or even the social co-participation. Broadly speaking, the decision making upon attending events depends on some restrictions such as time, location, category, popularity and which friend will attend. However, the existing techniques (e.g. collaborative filtering and content-based methods) cannot cope at all with the complex inherent nature of such decision. In addition, a recommender system is often application-specific, that is, to be tuned according to the item context (e.g. type, reasons to select an item, etc.). Another challenge in our work is that events often involve different topics (e.g. different genres in one musical concert). As a result, the user profile constructed based on the attended events may contain a wide variety of topics. This leads to topically diverse profile that may conceal the effective user interests.

To tackle these issues, we propose a hybrid recommender system based on Semantic Web technologies [97]. Our belief is that a structured representation presents one solution to cope with the complexity of event-specific characteristics. This modeling will ensure a more straightforward way to explore and reason over the data. It makes possible to ask complex queries, for example, to retrieve events involving the same artist within a specific geographical area. In addition, the semantic model empowers the enrichment of event descriptions with additional information from Linked Data. Such enrichment can provide valuable inputs for the content-based recommender system [47].

In the second step of our approach, we propose to quantify the user interests based on topic modeling technique. The objective is to detect the user propensity

towards specific topics. It will be integrated in the recommender system in order to control the impact caused by the diversity of a user profile. Finally, we exploit the collaborative participation assuming that the social information about "which friend will attend an event" plays an important role in decision making. In this work, we mainly investigate the extent to which the data enrichment, the social information and the user interests modeling can improve the system performance.

6.1 Content-based Recommendation using Linked Data

The principle of content-based (CB) recommendation is to suggest new items similar to those a user liked in the past. The similarity between items is computed based on the descriptive features of the item using a distance measure such as Cosine similarity, Pearson correlation and Latent Semantic Analysis [105]. The most common representation of the item is the keyword-based model, in which attributes are represented by weighted vectors of keywords usually computed by TF-IDF scheme (term frequency/inverse frequency). To build such a profile from unstructured data, feature extraction techniques are needed to shift the item description from the original representation to a structured form suitable for next processing (e.g. keyword vectors). This task becomes straightforward by the use of Semantic Web technologies. CB recommender systems can greatly benefit from the ease of ontology-enabled feature extraction, and the availability of Linked Data covering different domains to enrich the item profile. In the following, we explain how to compute the items similarity in Linked Data.

6.1.1 Items Similarity in Linked Data

In order to compute the similarity between items in Linked Data, we resolved to apply the approach proposed by Di Noia et al [47]. The key idea is that semantically similar items from RDF graph are the subject of two RDF triples having the same property and the same object (where a triple=<subject,property,object>). The intuition behind is that: *if two subjects are in the same relation to the same object, this is evidence that they may be similar subjects.* Technically, the approach is based on an adaptation of the classic Vector Space Model (VSM) [156], a well-known technique in Information Retrieval (IR). In this model, similarity between documents and queries is computed using their representative t-dimensional weighted vectors of discriminating terms. The application of VSM in RDF graph projects the Linked Data to 3-dimensional tensor where each slice represents an adjacency matrix corresponding to one property in the ontology. Indeed, the Linked Data network can be defined as a graph $G = (V, E)$ where V is a set of resources and E is the set of properties between resources in V. For each property p in the set E, the related adjacency matrix presents the linkage between the subjects (on the rows) and the objects (on the columns) from V via p. Then, a non null weight is assigned to each entry $X_{i,j,p}$ in

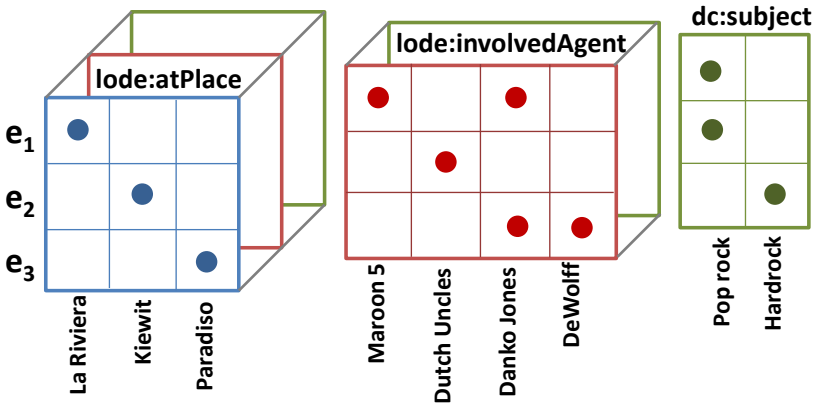

Figure 6.1: Tensor slices of some event properties (place, agent and subject)

the tensor for each existing triple $< i^{th}$ subject, p^{th} property, i^{th} object$>$. Figure 6.1 shows an example of tensor slices related to some properties, namely: `lode:atPlace`, `lode:involvedAgent` and `dc:subject`.

Assuming that the properties are semantically independent, we would be able to compute the similarity between events according to each property separately. The representation of an event e_i according to the property p is a t-dimensional vector indexing the terms/objects related to e_i via p. The TF-IDF weight of each object o is:

$$w_{o,i,p} = f_{o,i,p} \cdot log \left(\frac{N}{m_{o,p}} \right) \tag{6.1}$$

where $f_{o,i,p} = 1$ if a link exists between the node e_i and the object o via the property p, otherwise $f_{o,i,p} = 0$. N is the total number of events in the dataset, $m_{o,p}$ is the number of events linked to the object o via the property p. Then, the similarity between two events e_i and e_j according to the property p is computed using Cosine distance between their representative vectors as following:

$$sim^p(e_i, e_j) = \frac{\sum_{r=1}^{t} w_{r,i,p} \cdot w_{r,j,p}}{\sqrt{\sum_{r=1}^{t} w_{r,i,p}^2} \cdot \sqrt{\sum_{r=1}^{t} w_{r,j,p}^2}} \tag{6.2}$$

This approach can be applied to detect similarity between subjects or objects of RDF triples. It has been successfully used to recommend movies and to improve the quality of content-based system [47]. However, it is still limited when the adjacency matrix is very sparse such as the case of matrices associated with `lode:atPlace` and `lode:involvedAgent` properties. In fact, such predicates are characterized by the diversity of their object values, thus considered as discriminant properties. For instance, the t-dimensional vector related to `lode:atPlace` property has only one non-zero weight since an event is typically held at only one venue.

6.1.2 Similarity-based Interpolation

In order to mitigate the sparsity of the adjacency matrix, we propose to interpolate fictitious values based on the similarity between objects. Thus, we initially introduce a discriminability metric (i.e. discriminant power) to gain insight into the properties associated with highly sparse matrices. The metric is defined as following:

$$Discriminability(p) = \frac{\mid \{o \mid t =< s, p, o > \in G\} \mid}{\mid \ t =< s, p, o > \in G \mid} \tag{6.3}$$

where G is the RDF graph, t is the triple representing the link between the subject s and the object o via the property p. This formula quantifies the discriminability by the number of different object values on the target property. For instance, from a set of 1700 events (related to 10,323 agents, 627 places and 5,758 subjects), we found a discriminability score of 0.64 for the `lode:involvedAgent` and 0.45 for the `lode:atPlace`, while it is only equal to 0.10 for the `dc:subject` predicate. Furthermore, similar events are not necessary occurred at the same location or featuring the same performers. In order to reduce the discriminability impact, we interpolate fictitious weights in the adjacency matrix based on the similarity between objects as depicted in Figure 6.2.

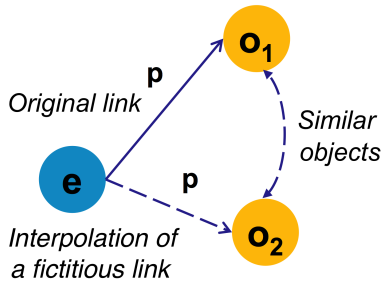

Figure 6.2: Similarity-based Interpolation

More precisely, if an object o_k is similar to another object o_h, and if both $f_{o_h,i,p} = 1$ and $f_{o_k,i,p} = 0$, then $f_{o_k,i,p} = sim(o_k, o_h)$. Note that $f_{o_k,i,p}$ reflects the strength of the fictitious link which associates the event e_i with the object o_k via the property p. If the object o_k is similar to more than one object originally linked to the event e_i, the weight $f_{o_k,i,p}$ will be equal to the highest similarity score. Thus, for each object o_k, the equation 6.1 becomes:

$$w_{o_k,i,p} = \max_{o_h \in H} sim(o_k, o_h) \ \cdot \ log \left(\frac{N}{m_{o_k,p}} \right) \tag{6.4}$$

where H is the set of objects originally linked to the event e_i. The intuition behind this formula is that: *if two subjects are in same relations to similar objects, this is evidence that they may be similar subjects.* We do not pay attention to how similarity

between objects is computed. In fact, this measure depends on the nature of the object itself and there exist several existing techniques that can be used. In our case, we exploit the similarity scores between agents (i.e. artists) provided by third party services such as Last.fm, and we compute the normalized geographical distance between venues.

6.2 Event Recommendation

Different from a classic item, events occur at a specific place and during a period of time to become worthless for recommendation. Moreover, while a classic item (e.g. movie, book) continuously receives useful feedback, an event has few rating due to its transiency. In our dataset, these ratings are represented by the binary user-event attendance matrix which has a sparsity rate equal to 98% (i.e. a set of users attend a very limited number of events). As a solution, one can address event recommendation using CB recommender system that exploits the matching of event attributes with the user profile. This perfectly complies with the constraints considered when it comes to decide whether or not to attend an event. Metadata such as distance, time, topics and artists are important and influential factors in such decision. Still, the CB recommendation might suggest items with a limited diversity and overlook the social information regarding the question "which friend is going?". To reduce this gap, we propose to enhance its performance by enriching the content using Linked Data, and by improving the detection of the user interests. Then, we incorporate the social information using Collaborative Filtering (CF) method, thus producing a hybrid recommendation.

6.2.1 Content-based Recommendation

The CB recommender system suggests future events similar to those a user has attended in the past. We assume that there is a sufficient number of past attended events in the user profile to avoid the *cold-start* problem[1], which is out of the scope of the present work. In order to predict the participation of the user u to the event e_i, we combine the similarity values between events as following:

$$rank_{cb}(u, e_i) = \frac{\sum_{e_j \in E_u} \sum_{p \in P} \alpha_p \; sim^p(e_i, e_j)}{\mid P \mid \; \cdot \mid E_u \mid} \tag{6.5}$$

where E_u is the set of past events attended by the user u, P is the set of properties shared between two events e_i and e_j, and α_p is the weight that reflects the contribution of the property p in the recommendation.

The properties selected to compute the similarity between events are those which are related to the location, subjects (tags) and involved agents (artists). In contrast,

[1]The problem to produce good recommendations for new users where nothing is known about their preferences

the temporal information is not considered in this work and left for future study. Our belief is that temporality could be harnessed to index the recent events in the user profile, thus reducing the computation. Still, there is a need to deeply investigate the impact that the reduction of the user profile has on the system performance.

Geographic Closeness

In recent research study, it has been shown that users generally tend to attend nearby entertainment events [143]. This fact makes the location a valuable feature in event recommendation. In our approach, we need to measure the similarity between events according to the `lode:atPlace` property. Thus, we normalize the distance between two locations using a specific threshold θ which needs to be determined. As the user home is missing in our data, we measure the distance between attended events for each user as depicted in Figure 6.3. Note that the attendance rate becomes extremely low from $\theta = 80$ Km. We consider that this value is the normalization threshold from which the similarity between events is equal to zero according to the `lode:atPlace` property.

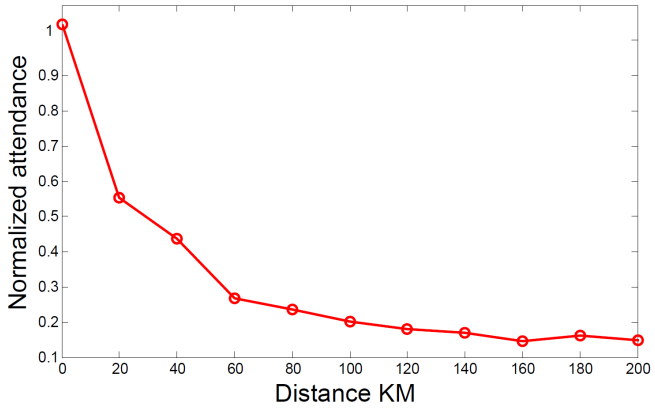

Figure 6.3: Normalized average attendance per distance

Enrichment with Linked Data

One method to enrich the item profile from Linked Data is to consume background information from DBpedia. The key advantage of DBpedia is the availability of semantically rich data in various domains. Using the mapping between EventMedia and DBpedia, we enrich the topics of an event using the DBpedia topics (e.g. genres) related to the involved artists. More precisely, we retrieve the categories associated with the property `dcterms:subject` of artists by simply querying the DBpedia SPARQL endpoint[2]. The reason behind our interest in DBpedia is that topics are accurately labeled and classified.

[2] http://dbpedia.org/sparql

6.2.2 User Interests Modeling

One fundamental goal in the recommender system is to suggest new items that best fit the user interests. In our case, this is particularly difficult to achieve due to the presence of topically diverse events. In fact, the real-world social events can be classified into large set of categories ranging from large festivals and conferences to small concerts and social gatherings. When attending an event, the user might be interested in a specific show or artist or might have broad interests. In consequence, relying on event similarity according to the `dc:subject` property can be influenced by the topical diversity of tags related to events in the user profile. To alleviate this impact, we leverage the Latent Dirichlet Allocation (LDA) [20] for detecting the relevant user interests as previously described in Section **??**.

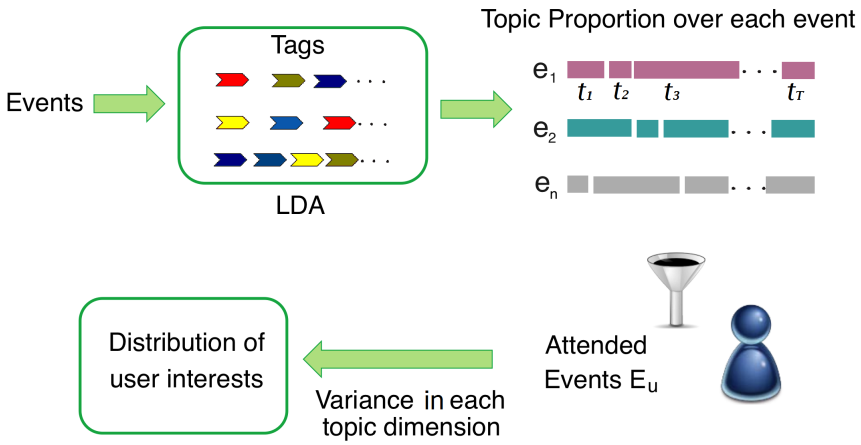

Figure 6.4: The pipeline of user Interests modeling

Figure 6.4 illustrates the pipeline of the user interests modeling. For each event e_i having a set of tags, 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 and Θ_i reflects the semantic categories of the event. Then, we compute the variance in each topic dimension t over all the events E attended by a user $\Theta^t = [\theta_1^t, \theta_2^t, ...\theta_E^t]$. The diversity score of each corresponding user is the mean of the variances of all the topics dimensions (mean of $\Theta^1, \Theta^2 ... \Theta^T$).

This approach as introduced by Wu et al. [181] has been originally designed to study the diverseness of individual tastes. But, we think that it is also helpful to detect user's propensity from a topically diverse profile. Indeed, events can be divided into two classes: those related to very few topics or those related to many topics. We consider that events in the first class are those which really exhibit the user interests. Using the variance, we are able to detect high proportions within topic dimension given that this dimension is likely to also contain low proportions (i.e. events are not regularly distributed over the topics).

As an example, Figure 6.5 shows the normalized diversity scores obtained from a

sample of 1,000 Last.fm users. In Figure 6.4(a), it is shown that most of diversity scores range from 0.3 to 0.5 indicating that users have relatively high interests in specific topics. The diversity scores near to 1 represent users having strong interests in very few topics such as the case of the user plotted in Figure 6.4(b). This user has a strong bias specifically towards the topic 9. Finally, the diversity scores close to zero generally represent the users associated with few attended events (i.e. the cold-start problem).

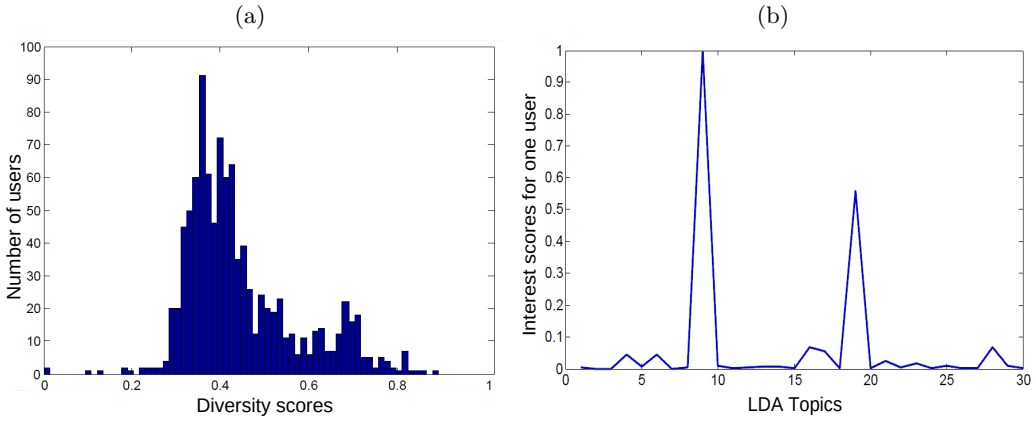

Figure 6.5: Distribution of topical diversity scores with T = 30: (a) for all the users; (b) for one specific user.

To take into account the effective user interests in a recommender system, we give emphasis to the events which are more likely to correspond to the user interests. We assign different weights β to the events included in the peaks of interest, and to those which are out of these peaks. These weights are then estimated using training methods. The content-based recommendation is extended as following:

$$rank_{cb++}(u, e_i) = \frac{\sum_{e_j \in E_u} \sum_{p \in P} \alpha_p \ \beta_p \ sim^p(e_i, e_j)}{\mid P \mid \ \cdot \mid E_u \mid} \tag{6.6}$$

where $\beta_p = 1$ if the property p is different from `dc:subject`, otherwise the $\beta_{subject}$ is an estimated value depending on whether the event e_j corresponds to the user interests or not.

6.2.3 Collaborative Filtering

A form of social interactions is the collaborative participation such as co-authoring a paper or co-attending an event. In [118], Liu et al. highlight the existence of an offline social network built from the co-attendance of social events. Accordingly, we consider that two users involved in the same event can potentially have a stronger tie than other users. Our assumption is that the more events in which users involve, the stronger is their tie. Thus, the co-attendance can be a clue to provide information

at first glance about which "friends" will attend an event. Moreover, our dataset contains the users' RSVP that express their intent to join social events, which can be exploited to predict unknown intents. However, unlike the traditional user-based collaborative filtering (CF), we decide to not only consider the similarity between users, but also the contribution of a group of friends. We define the following formula as the prediction that a user u_i will attend an event e based on the RSVP of his/her co-attendees (i.e. users who have attended past events with the user u_i):

$$rank_{cf}(u_i, e) = \frac{\sum_{j \in C} a_{i,j}}{\mid C \mid} \cdot \frac{\mid E_i \cap (\cup_{j \in C} E_j) \mid}{\mid E_i \mid} \tag{6.7}$$

where C is the set of co-attendees who will attend the event e, E_i is the set of attended events by the user u_i, and $a_{i,j}$ is the fraction of common events between the users u_i and u_j by the cardinality of E_j. Note that the weight $a_{i,j}$ reflects whether the most of events which are attended by the user u_j are also attended by the user u_i. The rationale behind this formula is two-fold: (1) in the first part, we consider the contribution of each co-attendee individually; (2) in the second part, we consider the co-attendees as a group of friends, and we assume that the more events they attended together with the user u_i, the more strongly is their relationship.

6.2.4 Hybrid Recommendation

To combine the predictions of both CB and CF recommender systems, we propose a weighted hybridization using a linear combination of predicted rank. Taking into account the user diversity and combining the equations (6.6) and (6.7), we proposed the following function:

$$rank(u, e) = rank_{cb++}(u, e) + \alpha_{cf}\ rank_{cf}(u, e) \tag{6.8}$$

where α_{cf} is the weight of CF method estimated in conjunction with the weights of CB features using optimization functions for training the system.

6.3 Evaluation

In this section, we carry out a set of experiments measuring the precision and recall metrics to assess the contribution of each step in our approach, and to evaluate the performance of our system compared with existing approaches.

6.3.1 Real-world Dataset

We use the EventMedia dataset and particularly the Last.fm directory which contains a large number of active users. Using SPARQL, we collected 2,436 events, 481 active users whose the attendance rates are within [15,50], generating 12,729 distinct consumption (i.e. user-event pairs). This set of events are related to 14,748 distinct

artists, 897 locations and 4265 tags (music domain). For the evaluation, we use a test set containing the most recent 30% of the consumption and a training test with the remaining 70% consumption. Then, we measure two metrics used in top-N recommendation task: Precision is the ratio of correctly recommended items and the length of the recommendation N; Recall is the ratio of correctly recommended items and the total number of future consumption. Precision and Recall are computed at different N values.

6.3.2 Learning Rank Weights

To learn the weights of our prediction function, we first test the linear regression with gradient descent that minimizes the least-squares cost function. Then, we use two evolutionary computation methods, namely the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO) motivated by their success in a wide range of tasks (details in Appendix B).

To apply GA in our approach, a chromosome is represented by a vector of the coefficients that need to be estimated. Each chromosome is then evaluated using a fitness function. This function aims to minimize the prediction error and thus maximize the precision of results. Table 6.1 shows the GA setting parameters.

Population size	Iterations	crossover	mutation
30	80	0.9	0.01

Table 6.1: Setting of GA parameters for event recommendation

As for PSO, a particle is represented by a vector of weights and the fitness function aims at maximizing the precision. Table 6.2 shows the PSO setting parameters.

Population size	Iterations	c_1	c_2	inertia
30	80	1.494	1.494	0.729

Table 6.2: Setting of PSO parameters for event recommendation

6.3.3 Experiments

First, we show in Table 6.3 the sparsity rates of similarity matrices according to each property. We can see the efficiency of our method to discover latent similarity between events especially for discriminant properties. This highlight the importance of the similarity-based interpolation and the enrichment using Linked Data. Unlike the keyword-based recommender systems, the interpolation is straightforward in our system thanks to the ontology-based data representation.

Task	location	agent	subject
(1)	0.9942	0.9174	0.3175
(2)	0.6854	0.7392	0.2843

Table 6.3: Sparsity rates of the similarity matrices before (1) and after (2) the similarity-based interpolation (for location and agent) and data enrichment with DBpedia (for subject)

Second, we assess the performance of the training methods to learn the coefficients α in the hybrid recommendation algorithm. Note that for this experiment, we do not include the user interests model and we set the $\beta_{subject}$ equal to 1. This experiment aims to rather compare the performance of optimization methods.

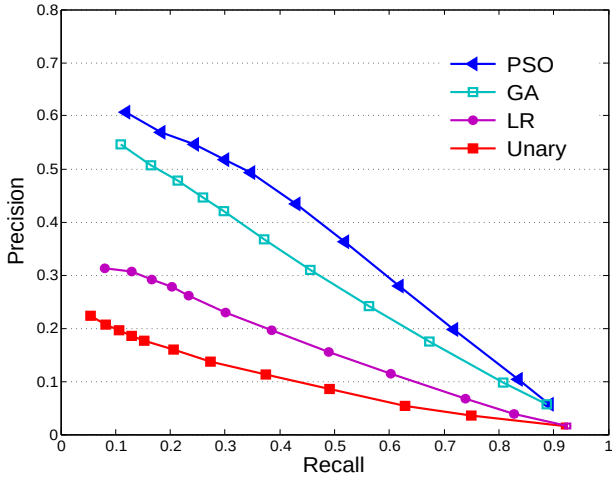

Figure 6.6: Recall and Precision using different approaches to estimate the vector α

Figure 6.6 shows the Precision and Recall curves. It is obvious that setting all the coefficients equal to 1 achieves the worst performance because there is no adaptive optimization. It is also shown that precision optimization methods (GA and PSO) yield considerably better results compared with error (RMSE) minimization method based in linear regression. This has been also proved in recent work [37] showing that methodologies based on error metrics do not necessarily improve the accuracy of top-N recommendation task. One given explanation is that the RMSE-oriented methods rely only on known ratings to train the system and do not consider the unrated items. Finally, Figure 6.6 highlights the better performance of PSO compared with GA algorithm. We observed a faster convergence to the optimal solution in PSO compared with GA which needs more iterations. This is due to the inherent behavior of PSO where the evolution is only guided by the best particle. In contrast, the GA evolution is guided by a group of solutions in which even weak candidates continue to survive after some iterations. In the following, we use the PSO algorithm to train the system.

To gain insight into the influence of the different steps in our approach, we examine the evolution of the system performance by incorporating in each experiment (by order) the enrichment with DBpedia, the user interests model and the collaborative filtering. Results are illustrated in Figure 6.7. We can observe that enriching data with DBpedia slightly improves both precision and recall. Indeed, introducing more coherent and qualitative data is one solution to reduce the noise that can be found in the collective knowledge of crowd tagging (e.g. Last.fm tags). Then, the user interests modeling also enhances the system performance. For this experiment, we fix the coefficients α obtained with PSO. Then, we train the system to compute the coefficient $\beta_{subject}$ which depends on the peaks of the user interests. As a result, we obtain $\beta_{subject}$ equal to 0.4 when the event is not included in an interest peak, and $\beta_{subject}$ equal to 1.6 (4 times more) otherwise. This proves the importance to clearly discern the user interests when the user profile contains diverse topics. Finally, combining these results with the collaborative filtering notably increases the recommendation accuracy. Our belief is such improvement is perfectly tangible with the use of a real-world dataset. According to the user centered study presented by Fialho et al. [54], social information such as people and friends who are attending an event has strong priority and influence on decision making.

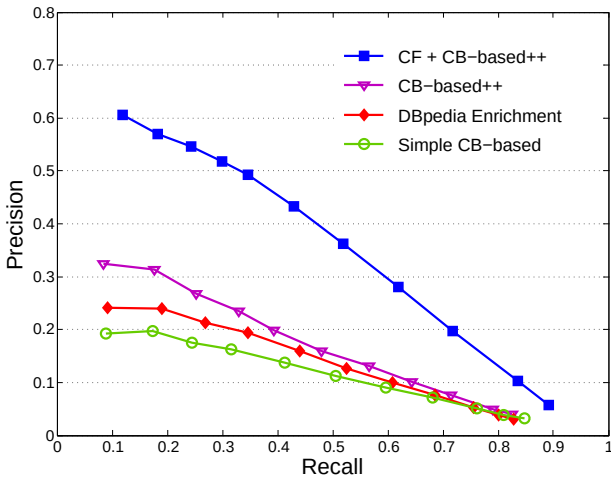

Figure 6.7: Evolution of the recommendation accuracy by incorporating the DBpedia enrichment, user diversity (CB-based++) and collaborative filtering (CF)

Lastly, we assess the extent to which a hybrid event recommendation outperforms the existing collaborative filtering based on matrix factorization to detect latent factors from the user-item matrix. We compare our system with the traditional user-based CF and the Probability based Extended Profile Filtering (UBExtended) proposed by Pessemier et al. [141] to recommend events. This method employs a cascade of two user-based CF systems aiming to recommend the most consumed (i.e. popular) events. The rationale behind is that the probability to consume an event is

proportional to the current popularity of the event (i.e. has attracted many users). The comparison results are depicted in Figure 6.8. It is shown that the UBExtended method outperforms the user-based CF algorithm. Still, the hybrid recommendation exhibits the best results in terms of precision and recall. This is due to the benefits of hybridization as has been highlighted in other research studies [148].

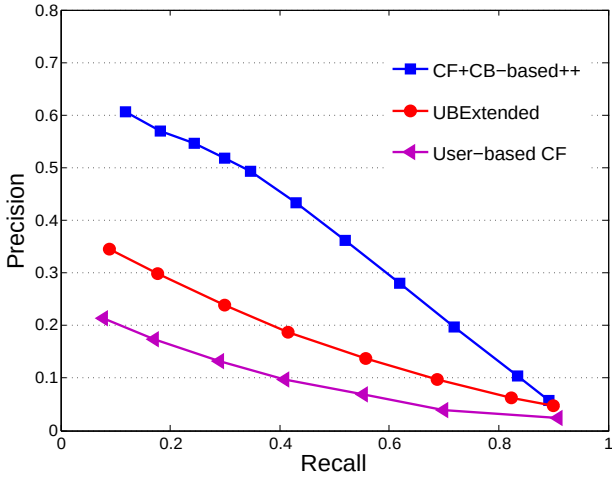

Figure 6.8: Comparison of hybrid event recommendation with pure CF algorithms

6.4 Related Work

In the research area of recommender systems, many approaches have been proposed to recommend movies, but few are the studies that deal with event recommendation. Events are particularly hard to recommend due to their short life time and the system often suffers from high sparsity of rating data. Some works have been proposed to overcome these issues and improve the recommendation accuracy. Cornelis et al. [36] built a hybrid approach within a fuzzy relational framework that reflects the uncertain information in the domain. The rationale behind is to recommend future events similar to those like-minded users have liked in the past. However, this framework was not evaluated and there is no clear insight about its performance. Minkov et al. [131] followed the same rationale and proposed a low rank collaborative method to predict the rating of future events. They highlight the performance of the collaborative filtering over the content-based system. Still, their approach was more tailored to recommend scientific talks in the same building and there is no consideration of the geographical constraint. Some other systems have been developed such as "Pittcult" [108] and "Eventer" [87] that position the user within a social network and leverage the trustworthiness between users. Such feature is valuable for recommendation, but it is not available in many systems. Finally, a user centered evaluation [51] showed that the straightforward combination of CF and CB recommendations outperforms both individual algorithms on almost qualitative metrics such as accuracy,

novelty, diversity, satisfaction and trust. Another interesting related works are the recent studies that harness the power of Linked Data in recommender systems. In [47], Di Noia et al. use the Linked Data as the only background knowledge to recommend movies. They highlight the performance of the system that exploits ontology-based data representation compared with the keyword-based representation. Still, there is no deep exploitation of the latent similarity that may exist between movie attributes (e.g. two similar actors).

6.5 Conclusion

In this chapter, we presented an approach for event recommendation combining both CB and CF advantages, and using Linked Data to enrich the event profile. In addition, we proposed an approach to model the user interests and to overcome the topical diversity in the user profile. The evaluation particularly highlights the importance of social information and the user diversity model to enhance the system performance. In the future, we plan to take into account other significant features such as event popularity and temporal indexing of recent consumption.

Chapter 7

Topical Community Detection in Event-based Social Network

Websites such as Lanyrd, Last.fm, Flickr and Twitter host an ever increasing amount of event-centric knowledge maintained by rich social interactions. In particular, there exists two types of *Event-based Social Networks* (EBSN). The former is represented by the typical online activities such as sharing media and exchanging thoughts. The latter captures the face-to-face social interactions reflecting the offline co-participation in the same events. For example, in the academic conferences, researchers interact with other community members with whom they share common research background [114]. In other words, EBSN is a heterogeneous social network underlying the co-existence of both online and offline social links [118]. Meanwhile, the information about these social interactions are spread over multiple websites. For example, people tend to mostly use media platforms (Flickr, Twitter) to share photos and thoughts about events, whereas they express their intent to attend events (RSVP) in online event directories (Eventful, Last.fm). Exploiting the overlap of these distributed websites is a key advantage to analyze the social networks.

Community detection is considered as a major topic for analyzing social networks and has recently received a great attention. It aims to uncover the substructures within a network revealing which users are likely to have common interests, occupations and social properties. The information about the underlying communities can be of a great benefit for many tasks such as information diffusion and personalization. For instance, a personalization system can be based on user's community to gain more knowledge about his/her behavior [159, 140]. It has been also proved that the substructures within a network provide new powerful means of recommendation and collaborative filtering [99, 140].

7.1 Background

Broadly speaking, detecting communities is dividing the vertices into groups such that there is a higher density of links within groups than between them [33]. To achieve this, most of existing methods focus on network topology and structural properties. They assume that the interaction strength of users is the reflection of their proximity. However, communities detected by link-based methods often represent users having different interests since no consideration of the topical dimension was made. It is

difficult to interpret the relationships of users grouped within such communities [38, 185]. This problem becomes more important when users interact with objects related to diverse topics. Therefore, merging the semantic information with the linkage structure is essential in order to detect meaningful and interpretable communities.

In EBSN, it is ideal to analyze the rich content about users and events in order to discover semantically coherent communities. Moreover, a person is naturally interested in many events which may be associated with multiple topics. It is thus more reasonable to divide users into overlapping topical groups instead of disjoint ones. Still, communities produced by topic-based methods may contain weakly connected users. They do not consider the relationships between users which results in significant loss of social information. An efficient community detection algorithm should cluster individuals who are closely connected and share common topics.

In this chapter, we propose a novel approach to detect topical communities (i.e group of users sharing a common topic) by combining event clustering with link analysis. First, we compute the similarity of events based on the social information and the content attributes. Then, we use a hierarchical clustering to group events into different topics. Finally, a link-based function is defined to determine the effective user attachment to each community.

7.2 Related Work

Community detection has attracted attention in recent years leading to several interesting studies. Most of existing works attempt to detect disjoint communities by optimizing different link-based objectives. One popular example is the modularity optimization [33, 138] used to maximize the connectivity between nodes within one community and minimize the connectivity between groups. Another example is the minimization of a defined cut function in spectral methods [179]. These works mostly focused on structural properties and linkage patterns of people and they have been successfully used in some applications. However, they generally produce communities associated with different semantic topics.

To overcome the limitation of link-based methods, some studies attempt to exploit topic modeling techniques such as pLSA [75], LDA [20] and AT (Author-Topic Model) [164] used to detect topical communities. For example, the work in [113] made an analogy between the LDA document-topic-word and the user-topic-websites. The idea behind is that users sharing similar online access pattern tend to belong to the same topical group. This method primarily relies on the link information in a social graph, and it is only efficient when regular interaction patterns can be detected. Another technique called Community-User-Topic (CUT) [186] extends the LDA model to detect communities using the semantics of content. As a result, communities are represented as random mixtures over users who are associated with a topical distribution. This method does not consider the link information assuming that community

members only share common topics. Obviously, both methods can not be applied in real-world social networks where users' memberships are conditioned on their social relationships as well as their shared interests [185].

Recently, some works start to investigate the combination of both content and link information. For example, the generative Bayesian model (Topic User Recipient Community Model) presented in [153] combines discussed topics, interaction pattern and network topology to detect topical communities. In [185], Zhao et al. proposed an approach based on a modified k-means algorithm (EWKM-Entropy Weighting K-means) to partition social objects (e.g. mails, events, etc). into topical clusters. Each topical cluster contains members who interacted with the associated social objects. Then, a modularity maximization method is employed in each topical cluster to detect strongly connected communities. In our work, we made analogy between these social objects and events and we extensively compare our algorithm with this approach (called EWKM-based method in the following).

The last concern in related work is the research on discovering communities in EBSN. Liu et al. [118] addressed the problem of community detection in heterogeneous network. Their approach is based on an extended Fiedler method to consider both the online and offline social interactions. This method seems efficient to detect cohesive communities, but it is still a link-based method and no interpretation of detected communities was made. In [114], the Event-based Community Detection (ECODE) algorithm enriched the network with virtual links based on content-based users' similarity. Virtual links aim to enhance connectivity among individuals sharing common topics (e.g. content). A hierarchical clustering is then used to group events based on their physical and virtual similarity. In the same context, Wang et al. [176], proposed a community detection approach in location-based social network (LBSN). Their approach exploits different features such as user social similarity and venue-user similarity, and uses an edge-centric co-clustering which simultaneously discovers overlapping groups of venues and that of users. To sum up, these different studies provide important insight into detecting communities in EBSN. However, none of them aims to maximize both connectivity strength and topical purity within communities.

7.3 Event-based Social Network

In this section, we describe how to construct an event-based social network using offline and online interactions (Section 7.3.1) and we highlight some of their interesting properties (Sections 7.3.2 and 7.3.3).

7.3.1 EBSN Definition

Based on user activities in social services, we define the following EBSNs making difference between online and offline networks. Slightly different from the definition

given by Liu et al. [118], we consider that the online EBSN is constructed by solely capturing the online interactions such as sharing comments and photos about events. This online EBSN is different from the online "friendship" social network that may exist in some services as defined in [118]. Similarly, the offline EBSN is constructed by considering the physical co-participation in social events.

- **Event Directory**
  - **Last.fm EBSN.** In Last.fm, there are two networks: the online EBSN is built based on the online co-commenting of social events, whereas the offline EBSN is based on the explicit RSVP provided by users.
- **Media Directories**
  - **Flickr EBSN.** Flickr is one of the most important online photo sharing websites. Thus, we exploit the activity of co-sharing photos related to the same events to build an online media EBSN.
  - **Twitter (Lanyrd) EBSN** Twitter is a popular micro-blogging service, and it is by far the most used back-channel for commenting scientific conferences [90]. Similarly, we exploit the co-commenting activity about the same conferences to build another online media EBSN.

7.3.2 Spatial Aspect of Social Interactions

In the following, we investigate how far from their homes people interact within the offline and online EBSNs. Therefore, we compare the geographical distance between an event location and the user's home. However, as the user's home location is not explicitly provided by Last.fm, we infer it using the average of most frequent positions of attended events. Results are depicted in Figure 7.1 based on a random set of events and their associated users.

We observe that 95% of users' activities in offline network are within 100 km. This rate slightly decreases in online Last.fm EBSN indicating that people tend to also comment nearby events. This aspect has already been proved in an existing study [118] showing that users' activities in EBSNs are much more location constrained compared with location-based social network. In contrast, the online interactions in media-based EBSNs seem to be less conditioned on event location. The reason behind can be two-fold: (1) the nature of sharing activity which is more present in media platforms than event directories, and the users are generally non-uniformly spread; (2) the type of events indicating that people tend to travel far from their home for business purpose (conference) rather than for entertainment activity (musical concert).

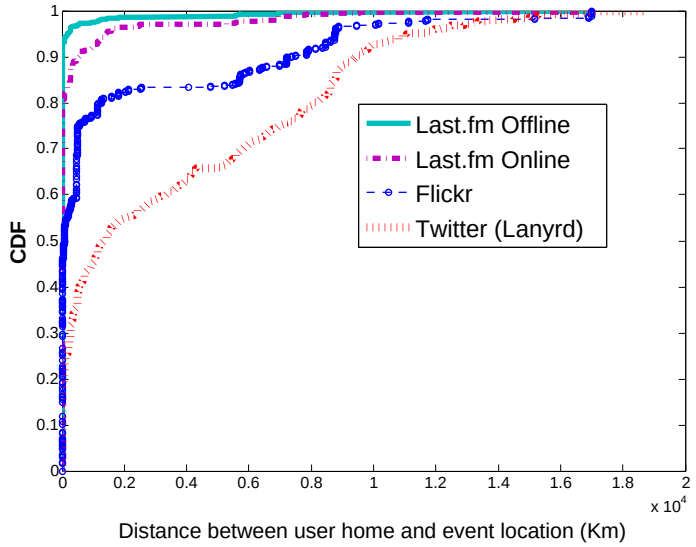

Figure 7.1: Locality of user activities in offline and online EBSNs

Based on these findings, we decided to perform community detection using conferences from different cities in Lanyrd, whereas we only focus on a specific geographical location in Last.fm.

7.3.3 User Participation

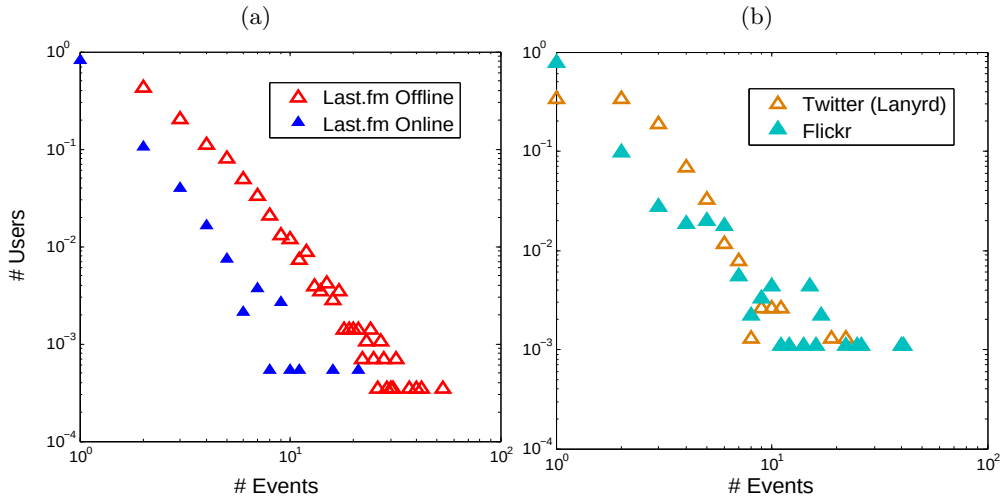

Figure 7.2: Number of participants per event in (a) Last.fm offline and online EBSN and (b) Flickr and Twitter online EBSN

To gain insight into some EBSN properties, we study the user participation behavior. As shown in Figure 7.2, the results resemble a power-law distribution indicating that most of users are associated with few events. Similar results have been

highlighted in other works studying the event attendance behavior [118, 65]. In particular, there are 81% of users who are associated with only one event in Last.fm online EBSN, and 76% of users sharing photos of only one event in Flickr EBSN (this can be drawn from Table 3.1 if we compare the number of media shared with the number of users). During the evaluation, we will show the influence of the user participation distribution on community detection.

7.4 Topical Community Detection

In this section, we first describe our graph model (Section 7.4.1). Then, we present our approach proposed to detect topical communities.

7.4.1 Graph Modeling

Taking into account the users, the events and their related attributes, we consider the fourth-tuple graph $G =< U, S, T, E >$ for both online and offline EBSN where U is the set of users, S is the set of social events which are in turn associated with a set of tags, and finally E is the set of undirected edges. E contains two kinds of links $E = E_{US} \cup E_{UU}$:

- E_{US} denotes the links between users and social events, formalized as $E_{US} = \{(u, s)|u \in U, s \in S\}$
- E_{UU} is the set of links between users (i.e. a link represents the co-participation in same social event), formalized as $E_{UU} = \{(u_i, u_j)|u_i \in U, u_j \in U\}$.

In this graph, each user can be represented as a vector of events, and each event can be represented as a vector of users. Similar way is applied using the event-tag relationship. We exploit these representations to compute the similarity of events which will be used for detecting communities.

7.4.2 The Proposed Approach

In our approach, we follow the same rationale as the EWKM-based method proposed by Zhao et al. [185]. The idea behind is to group together the social objects which share the same topic, and . Then, users within each cluster are divided into sub groups using the modularity maximization. Instead of performing two-step clustering, we propose one step clustering taking into account both the link and content information.

7.4.2.1 Similarity Computation

In EBSN, the overlapping communities of users who share common interests can be detected by clustering similar events together [114]. Given the large number of users, we assume that event-based clustering have a less computational time compared

with user-based clustering. Still, the event similarity should reflect both the link and content information in order to discover topical communities. To solve this, we use the notion of *Homophily* which is observed in many social networks [125]. Homophily refers to the tendency of persons to be associated with other persons that share similar characteristics. In other words, users involved in same events have a higher likelihood to share similar interests and get connected. Similarly, tags associated with same events are more likely to be topically similar. This implies that similar events are sharing both like-minded users and semantically similar tags. Thus, we cluster events based on their similarity both in the user space and in the semantic space.

In the event-user network, events can be represented as a vector of users, and users can also be viewed as a vector of events. To reduce the dimension of the event-user matrix, we need to represent events in a latent user space using an orthogonal basis. Singular Value Decomposition (SVD) is one popular technique employed to obtain such basis. Given a matrix A, the singular value decomposition is the product $U\Sigma V^T$ where U and V are the left and right singular vectors and Σ is the diagonal matrix of singular values. Event vectors in the latent user space is represented by the matrix $\tilde{A}$ as follows:

$$\tilde{A} = U\Sigma \Rightarrow \tilde{A} = AV \tag{7.1}$$

In order to detect similar events that share like-minded users, we leverage the spectral co-clustering [46] indicating that only the top singular vectors, except of the first one, contain partition information. The algorithm first normalizes the event-user matrix as follows:

$$A_n = D_1^{-1/2} A D_2^{-1/2} \tag{7.2}$$

where the entries of the diagonal matrices D_1 and D_2 are respectively the event degrees and the user degrees (i.e. a degree is the number of connections the node has to other nodes). Then, applying SVD on A_n gives $A_n = U_n \Sigma_n V_n^T$. Only the top-k singular vectors (except of the first one) are selected from $V_n = (v_1, v_3, ..., v_n)$ to form the matrix $V_n' = (v_2, v_3, ..., v_m)$ where $m \ll n$. Finally, the event representation in the user latent space is shown in Equation 7.3.

$$\tilde{A} = D_1^{-1/2} A_n V_n' \tag{7.3}$$

Similarly, we represent events in the latent semantic space applying this method on the event-tag network. Recent experiments in text corpus suggests that the dimension m of V_n' depends on the corpus size and it was set between 50 and 1000 [104]. Indeed, small value of m is advantageous to remove noisy information. In our case, we set m equal to 200 for the user space and equal to 50 for the semantic space (there are more users than tags in our dataset). Afterwards, we use Cosine distance to compute

the events similarity S_u in the latent user space and S_t in the latent semantic space. Finally, we combine the similarities as follows:

$$S_{sim} = \alpha \; S_u + (1 - \alpha) \; S_t \tag{7.4}$$

where α is the parameter that controls the balance between user-oriented similarity and tag-oriented similarity. In this approach, the pair-wise computation using Cosine distance can be reduced by selecting candidate solutions that only index the potentially similar events. Intuitively, these solutions are the events that share in common a minimum number of tags or users with the original event. Variants techniques can be used such as the Locality Sensitive Hashing (LSH) [59] or its variants (e.g. MultiProbe LSH [120]) which are popular high-dimensional similarity search methods. In ECODE algorithm [114], it has been shown that the candidate selection was efficient to save a significant amount of computational time without affecting the communities detected. Although it can be easily applied, candidate selection is not considered in the present work since we deal with small datasets.

7.4.2.2 Hierarchical Clustering

Inspired by the ECODE algorithm [114] described in related work (Section 7.2), a hierarchical agglomerative clustering is used to group similar events in terms of correlated users and tags. Agglomerative clustering begins by assigning each data item to its own individual cluster. The two most similar clusters are merged together into a single cluster. This step is repeated until all the items are grouped into a single cluster, thus forming a hierarchy (i.e. tree).

Algorithm 1 Agglomerative clustering of similar events

S: set of social events $s_1, s_2 ... s_i$
T: number of topics
S_{sim}: event similarity matrix
while Community Size>T **and** *SemQ* function increases **do**
  Merge the most similar events s_i and s_j into a new event s_{new}
  for each event $s_k \in$ S (or candidate set) **do**
    $S_{sim}(s_{new}, s_k)$ = average($S_{sim}(s_{new}, s_i)$+$S_{sim}(s_{new}, s_j)$)
  end for
  Compute *SemQ*
end while

As outlined in Algorithm 1, the most similar events s_i and s_j are clustered together forming a new event s_{new}. Then, we compute the similarities between s_{new} and each event in the dataset or in the candidate set (if the candidate selection is considered). Finally, the clustering stops when there is no significant increase of the quality function. This approach is advantageous compared with other algorithms such as k-means since the predefined number of clusters is not required.

To produce topical clusters, the quality function of the tree follows the same rationale than Newman modularity but applied on the semantic space. Indeed, our goal is to maximize the intra-similarities and minimize the inter-similarities in the semantic space. Thus, we define a novel function called *semantic modularity*. To formalize this function, we use the events similarity S_t computed in the latent semantic space in order to compute the intra-similarities (IntraSem in Equation 7.5) and inter-similarities (InterSem in Equation 7.6) as following:

$$IntraSem = \frac{1}{|C|} \sum_{C_k \in C} \frac{\sum_{i,j \in C_k, iC_ij} S_t(i,j)}{|C_k|(|C_k|-1)} \tag{7.5}$$

$$InterSem = \frac{1}{|C|} \sum_{i \in C_i} \frac{\sum_{j \in C_j, C_i \neq C_j} S_t(i,j)^2}{M} \tag{7.6}$$

where C is the set of discovered clusters, and M is the number of comparisons made in inter-similarities. Finally, the semantic modularity $SemQ$ is defined as following:

$$SemQ = IntraSem - InterSem \tag{7.7}$$

Note that the maximal $SemQ$ provides the topical clusters of events and stop the clustering process. In meanwhile, each detected cluster keeps in mind a minimal knowledge about the link information held by the event similarity in the user space, which makes our approach different from the EWKM-based method [185].

7.4.2.3 User Assignment

The last step of our approach is to group together users associated with each cluster by simply using the user-event links. As the user may participate in many events, we can generate overlapping topical communities. However, a user may be weakly involved in one topical cluster that not really reflects his/her interests. To address this problem, we propose to discover the effective user's memberships by computing his/her assignment scores. If the user u_i is a member of the community C_i, the assignment function is defined as follows:

$$AS(u_i, C_i) = \frac{D_c(u_i)}{D(u_i)} \tag{7.8}$$

where $D_c(u_i)$ is the degree of the user u_i within the community C_i (i.e. number of links of the user with other community members), and $D(u_i)$ is the global u_i's degree. The user's membership to one community is determined if the related assignment score is higher than the average of non-zeros scores over all communities. Note that the user assignment method based on Equation 7.8 may convert a cluster to an empty one. We believe that the removal of these empty clusters is reasonable since they represent groups of very weakly connected users.

7.5 Evaluation

This section presents the evaluation of the proposed community detection approach applied on real-world datasets. We first describe these datasets followed by the description of the performance metrics and the obtained results.

7.5.1 Experimental Datasets

Based on the definition of online and offline EBSNs, we use the following datasets placed online[1] (some statistics are shown in Table 7.1).

- **Entertainment (Last.fm and Flickr):** We previously demonstrated that a very high fraction of social interactions for entertainment purpose exist between geographically close friends. Hence, we focus our analysis on events located in one city. The capital "London" has been selected since it exhibits a significant number of users and events compared with other cities in EventMedia. Operationally, we query the EventMedia SPARQL endpoint to retrieve data, and we crawl additional metadata using the REST API of Last.fm and Flickr. Then, we pre-process the dataset as follows: First, we remove the tags associated with very low frequency (less than 5) to reduce the topical noise, and we only keep the events which are associated with the frequent tags (musical genres). Second, we remove the singletons of event-user pairs where the event has only one participant, and this participant is associated with only one event. We retrieve the events happened in 2012 and 2013 (associated with media) and we obtain the following EBSNs: (1) an offline Last.fm EBSN containing 915 events, 2847 users and 272 tags; (2) The associated online Last.fm EBSN contains 470 events (among 915 events), 1729 users and 248 tags (among 272 tags); (3) The associated online Flickr EBSN contains 375 events, 868 users and 221 tags. Note that the removal of singletons event-user pairs has significantly reduced the size of the online Last.fm and Flickr EBSNs indicating that users' activities in those networks are more sporadic and mostly present individual behaviors.

- **Conference (Lanyrd and Twitter)** Similarly, we use SPARQL queries to retrieve data from EventMedia along with the Twitter API for additional information (e.g. user's home location). Note that Lanyrd also provides details about the conference attendees, but this information was missing in EventMedia at the time of writing. Thus, we plan to further enrich our dataset and we left the analysis of offline Lanyrd EBSN for future study. Then, we pre-process the data retrieved as follows: As no tags were associated with events, we automatically process the conference description (tokenization, stop-words removal, etc.). However, this method produced very noisy tags because some

[1] http://www.eurecom.fr/~khrouf/esbn

conferences are vaguely described (e.g *The World is Changing, Is Your Company on Board?*). We also attempt to automatically process the tweets. Still, many tags do not really reflect what is the conference about due to the presence of several noisy tweets (e.g. personal status updates, opinions, etc.). As an alternative solution, we manually label the conferences descriptions by selecting the most representative keywords. Due to the manual effort, we only keep the interesting conferences which are related with very active users. Finally, we obtain an online EBSN which contains 275 events, 768 Twitter users and 166 tags. Note that there is a small set of events compared with Last.fm EBSN due to the high selectivity followed.

	Edges	Density	ClustCoeff
Last.fm Offline	95897	0.0237	0.1144
Last.fm Online	9936	0.0067	0.398
Flickr Online	7071	0.0188	0.2624
Twitter Online	14237	0.0483	0.4852

Table 7.1: Some statistics about the datasets

7.5.2 Topic Modeling

In order to assess the topical purity in each cluster, we first need to detect the set of topics in each dataset. Thus, we decided to employ LDA [20], a popular topic modeling technique where we consider the events as documents. The use of LDA has led to coherent topics in Lanyrd dataset, but slightly ambiguous topics in Last.fm datasets. We explain this difference by the manual labeling in Lanyrd dataset where we carefully select qualitative tags. In contrast, Last.fm contains crowdsourced tags generated without moderator oversight and known to be less accurate. Moreover, the musical concerts may feature many artists related to different genres (i.e topics) or only one genre, making more difficult to detect co-occurrences. The conferences, on the other hand, often target only one major topic (e.g. Semantic Web).

To solve the topic modeling in Last.fm, we resolved to exploit the classification of musical genres that may help detect the fuzzy similarity between them (e.g. death metal and deathcore). More precisely, we leverage the existing SKOS taxonomy[2] in DBpedia using the generalization relations such as `skos:broader` and `skos:narrower`. Tables 7.2 and 7.3 show few examples of topics detected respectively in Lanyrd and Last.fm. Note that we obtained 24 topics in Last.fm consisting of high-level musical genres, and 30 topics in Lanyrd where the optimal number of topics is determined based on the approach proposed by Griffiths et al. [61]. Finally,

[2] http://dbpedia.org/page/Category:Musical_subgenres_by_genre

Figure 7.3 shows that many conferences have at most two topics, while this number slightly increases in musical events.

Topic	Example of Lanyrd Tags
Education	learning, education, teaching, technology
programming	programming, language, python, library
Innovation	creativity, technology, business, future
Application	mobile, application, web

Table 7.2: Example of topics detected in Lanyrd

Topic	Example of Last.fm Tags
Heavy metal	metal alternative, progressive metal...
Pop	synthpop, powerpop, pop punk...
Electronic	indietronica, synthpop, folktronica...
Rock	hard rock, alternative rock, glam rock...

Table 7.3: Example of topics detected in Last.fm

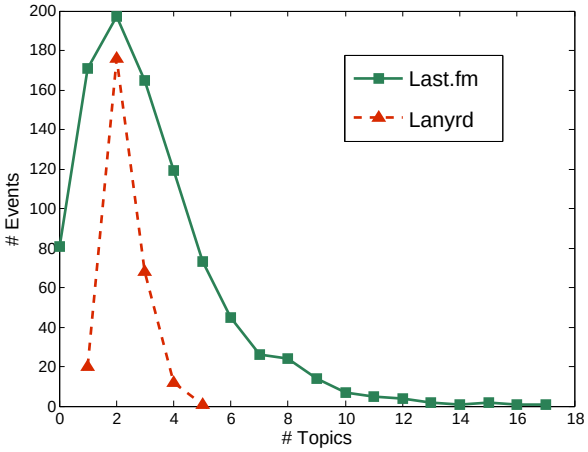

Figure 7.3: Histogram of the number of topics per event

7.5.3 Evaluation Metrics

To evaluate our approach, the performance metric should take into account the combination of both content and link information. We adopt the $PurQ_\beta$ metric introduced by Zhao et al. [185]. It has been inspired by the F-score measure that considers both the precision and the recall metrics. $PurQ_\beta$ considers both the topical purity and the members connectivity. First, we introduce the function that measures the topical purity in each cluster as following:

$$Purity_i = max_j \left(\frac{n_{ij}}{n_i} \right) \tag{7.9}$$

where n_{ij} is the number of tags belonging to topic j and cluster i, and n_i is the number of tags in the cluster i. The final score of $Purity$ is the overall average of all the purity scores. Yet, we observed during the experiments that Purity does not effectively reflect the presence of clusters having low topical purity. Hence, we decided to also examine the F_{purity} which is the faction of clusters having $Purity_i$ higher or equal than the average $Purity$. Finally, the metric $PurQ_\beta$ combining the content and link information is:

$$PurQ_\beta = \frac{(1+\beta^2)(Purity \cdot Q)}{\beta^2 Purity + Q} \tag{7.10}$$

where Q is the Newman modularity [138] used to evaluate the goodness of a partition, ensuring that there are many edges within communities and only a few between them. Then, the parameter β is used to adjust the weight of $Purity$ and Q. $\beta = 0.5$ means that $PurQ_\beta$ puts more emphasis on $Purity$ than Q. In contrast, $\beta = 2$ puts more emphasis on Q. The general behavior of communities is when $Purity$ increases, Q decreases, and vice versa.

7.5.4 Results

We first evaluate how the coefficient α in Equation 7.4 affects the performance of our approach. Figure 7.4 shows the evolution of the $Purity$ and the modularity Q when α increases. It can be seen that the modularity increases if a high weight is assigned to event similarity in the user space. However, the purity and the modularity do not evolve at the same scale. While Q slightly increases, $Purity$ drastically decreases. Thus, good values of $PurQ_\beta$ can be obtained when $\alpha \in$[0.1,0.5].

Then, we compare our approach with some related works: (1) Edge co-clustering (or EdgeCluster) inspired by the approach applied on the location-based social network and proposed in [176]. For this approach, we consider as features the user similarity in the event space and in the semantic space. Based on these features, Edge co-clustering uses k-means to cluster similar "user-event" edges. This method was evaluated only on two datasets as it requires a very large computation time; (2) ECODE algorithm which introduces the concept of content-based virtual links in the user-event graph and clusters together similar events sharing high physical and virtual links; (3) The popular Newman Modularity maximization (link-based method); (4) The EWKM-based method as detailed in related work (Section 7.2).

The comparison results are depicted in Figure 7.5. All these methods have nearly similar performance in Last.fm Online EBSN particularly when $\beta = 0.5$. Indeed, the communities detected within this network have very small sizes (e.g average size equal to 15 for the Modularity method) due to the extremely sporadic interactions. This is

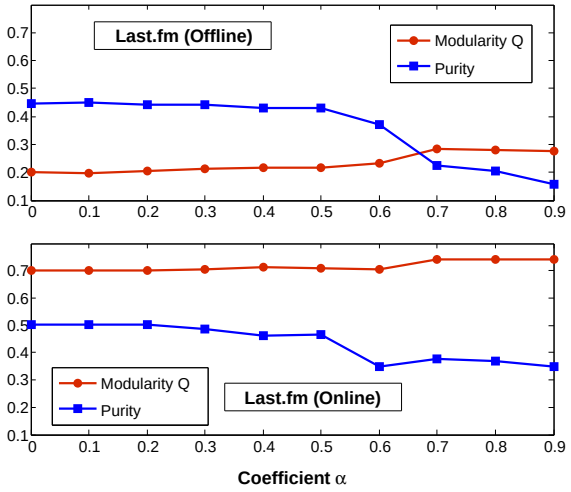

Figure 7.4: The evolution of Q and Purity with α

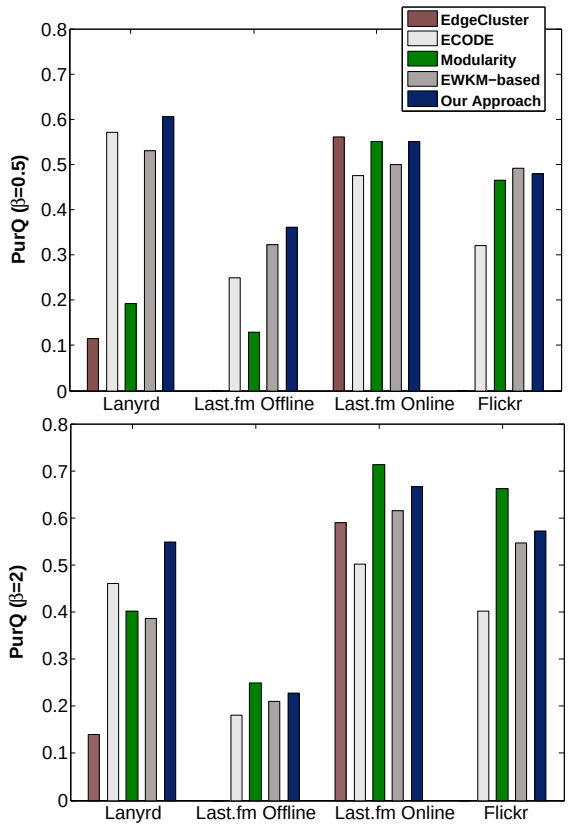

Figure 7.5: The performance comparison with $\beta = 0.1$ and $\beta = 2$ for different datasets

also explained by the low density link and the user participation behavior where 92% of users are associated at most with only two events. Hence, the link information was sufficient to obtain a good purity. This aspect slightly decreases in Flickr dataset where 78% of users are associated with at most two events. The Modularity method apparently achieves a good purity. However, the fraction F_{purity} is only equal to 0.6, a fair value compared with EWKM-based method and our approach where F_{purity} are respectively equal to 0.89 and 0.91. In Last.fm Offline and Twitter EBSNs, the Modularity method has a poor performance when $\beta = 0.5$. This can be explained by the network density which is high in those datasets compared with others. Moreover, the identified communities are very large. For example, we found an average size of 474.5 in the communities produced by the Modularity method in Last.fm offline EBSN. This indicates that users within this network are densely linked which can justify the low Q values produced by different approaches.

Evaluating the content-based methods, we note a better performance for ECODE in Twitter EBSN than in the other datasets. This is due to the addition of virtual links to the graph based on the content-similarity between users. However, the user profile in Last.fm is much more topically diverse than in Lanyrd which leads to ambiguous similar scores. In reality, the user may be interested in many musical concerts having different topics, whereas he has more restrictive "scientific" interests that mostly fit his/her expertise domain. We also observe a poor performance of the Edge co-clustering algorithm in Twitter EBSN because it is sensitive to the number of clusters that needs to be accurately determined. Finally, our approach achieves the best performance both when $\beta = 0.5$ and $\beta = 2$. Note that there is similar behavior between our method and the EWKM-based method. For instance, the average size of communities in Last.fm Offline EBSN is equal to 0.33 for EWKM-based, and 0.29 for our approach. However, the EWKM-based method is based on k-means clustering which is sensitive to the initial distribution of centroids, thus producing different results in each run. This problem is absent in our approach which is based on hierarchical clustering. From the computation point of view, we observe that all methods have nearly the same computational time except of the Edge co-clustering method. Finally, low purity values are observed in Last.fm Offline EBSN compared with Twitter EBSN. The reason of this lower performance is that the musical concerts are attached to much more topically diverse tags than the conferences in Lanyrd. In the following, we select the EWKM-based method to further evaluate our approach.

Coductance Comparison

It is difficult to construct a ground truth that represents the real communities within a network. Hence, we evaluate the proposed approach using the *Conductance* metric [111]. Conductance is a popular quality function assessing whether the detected communities are densely linked but weakly attached to the rest of the net-

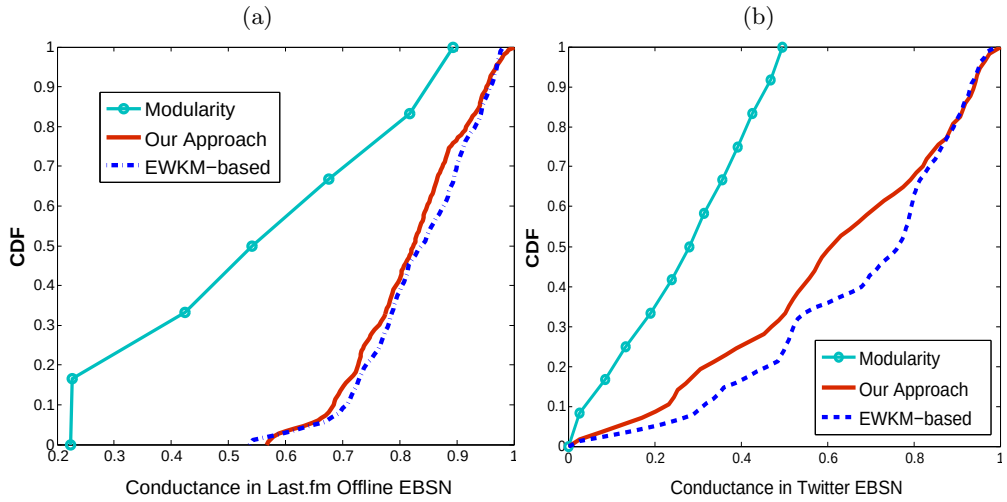

Figure 7.6: Conductance comparison in (a) Last.fm Offline EBSN and (b) Twitter EBSN

work. Note that this metric will evaluate our method from the link-based perspective. Lower conductance values mean better community structure. Figure 7.6 shows the cumulative distribution (CDF) of the conductance respectively in Twitter EBSN and Last.fm Offline EBSN. Our approach produces slightly more communities with lower conductance values especially in Twitter EBSN. The reason behind is the strategy to determine users' memberships based on their global degrees. We believe that the better performance in Twitter EBSN is due to its clustering coefficient which is larger than that of Last.fm Offline EBSN.

User Profiles Comparison

To evaluate our approach from the content-based perspective, one way is to compare the user profiles within one community. Hence, we retrieve the users' tags from each website and we only keep the frequent ones, thus creating a user profile. Cosine distance is then applied to compute the similarity between users' profiles. We consider that two users are similar when they have a Cosine distance above 0.3, a quite reasonable value considering the noisy tags. Figures 7.7 shows the CDF of the fraction of similar users within the same communities. It can be seen that our approach clustered more "topically" similar users than the EWKM-based method did.

We also examine the fraction of "friends" within each community. The friendship information was extracted using the online social networks that exist in Last.fm and Twitter detected by our approach. Results are shown in Table 7.4. We can see that a large fraction of friends were clustered in the same community by the Modularity method in Last.fm Offline EBSN compared with the other methods. This is also justified by the very high average size of communities detected which is equal to

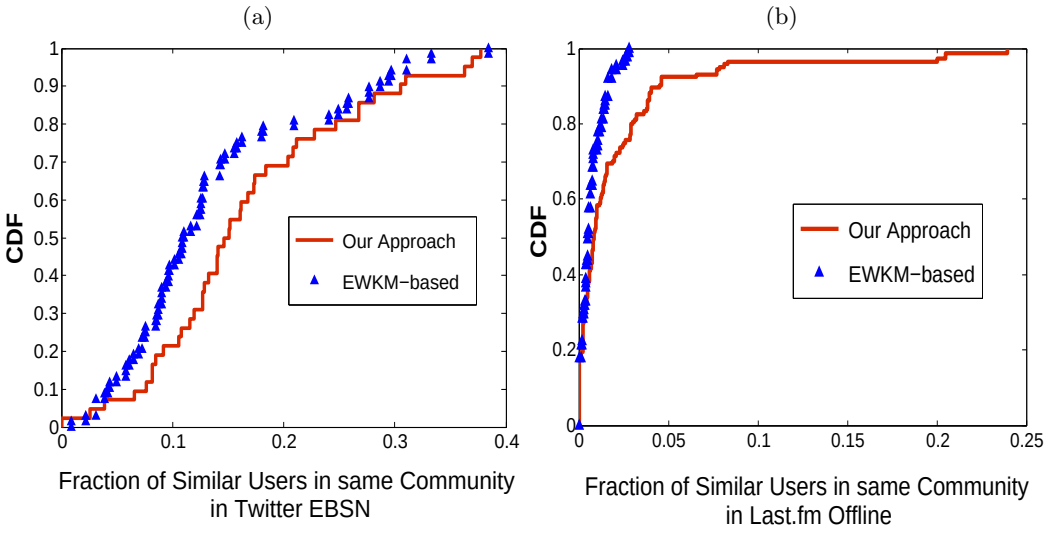

Figure 7.7: Comparison of user profiles in (a) Twitter EBSN and (b) Last.fm Offline EBSN

474.5. Moreover, it is clearly shown that the conference attendees having similar topical interests are more likely to be friends than the case of concert attendees.

Method	Twitter (Lanyrd)	Last.fm Offline
Modularity-based	0.72	0.69
EWKM-based	0.70	0.23
Our Approach	0.73	0.29

Table 7.4: Average fraction of friends within communities

Communities Overlap

Lastly, Figure 7.8 shows a tag cloud representing a sample of the most overlapping communities in Twitter EBSN. The link thickness exhibits the overlapping degree. It can be drawn that the main topic of these communities is the Web domain which is the interest of many users who share different "topical" expertise.

In Twitter EBSN, our approach detects 65 communities while the EWKM-based method produces 92 communities. Analyzing both community structures, it is found that our approach discovers fewer but more cohesive topical communities. We evaluate the cohesiveness using the popular Silhouette coefficient [149]. For instance, we have detect only one community about the topic "*user experience*" with a cohesion equal to 0.1. In contrast, 4 communities have been detected about this topic by the EWKM-based method including 2 singletons (i.e. community having one user) and having a cohesion equal to -0.3. This finding underlines the advantage of our

Figure 7.8: A sample of some overlapping communities in Twitter EBSN

approach to group together strongly linked users and to remove communities having weak connectivity.

7.6 Conclusion

Today's people use event and media websites to interact together either online by sharing comments and photos or offline by attending events. Thus, many social connections can be formed and strengthened during social events which can be considered as a basis to detect communities. In this chapter, we proposed a new approach to discover topical communities from event information. Taking into account both the content and the link information, we clustered events by maximizing a newly defined metric called *Semantic Modularity*. Then, the user membership to each cluster was determined by a link-based function based on the user's degree. A comparison with existing studies shows the efficiency of our approach to detect communities optimizing both users connectivity and topical purity. Results highlighted how people interact differently in offline and online EBSN and how these interactions depend on the event category (e.g conference, concert, etc.). For future work, we plan to combine both the offline and the online worlds to solve community detection in a heterogeneous network. We wish to assess the impact of such combination on the purity and the connectivity of topical communities.

Conclusion of Part II

This part has been devoted to put in use the Linked Data in event domain. Of a particular interest is the applications that handle better event presentation and discovery, as well as the personalization techniques.

Semantic Web applications have been developed to support end-users browsing and creating events. Overall, consuming Linked Data is advantageous to deliver enriched views of events, and to uncover interesting behavioral facts. Still, it was challenging to use conventional Web technologies on top of RDF data, a fact which reminds the trade-off between simplicity and expressivity.

Semantic Web technologies have been exploited to recommend events. Ontology-enabled feature extraction showed its ability to reduce the data sparsity, a problem from which suffers the traditional recommender systems. We highlighted that Linked Data is also beneficial to enrich data, thus improving the performance of our hybrid recommender system.

Finally, we proposed an approach to detect topical communities in event-based social network (EBSN) based on the content and link information. Linking events with media was particularly useful to construct EBSNs from media services. Evaluation shows how people interact differently from one service to another and depending on the event context.

Chapter 8

Conclusions and Future Perspectives

In this chapter, we summarize the major achievements of this thesis and we give an outlook on future perspectives.

8.1 Achievements

An ever increasing amount of information spread on the Web is centered on the notion of "event". Currently, most of companies that provide calendar of events such as Eventful, Last.fm and Lanyrd are using Web 2.0. They provide an environment where users can view and create an event, and locate events through keyword-based search and ranked results. Such design as unconnected data silos is, however, different from the conception to which aim the "Web of events". There is no support to handle the natural relationships that link events at different levels (e.g. similarity) or to link events with their experiential attributes such as discussions and captured media. Indeed, associating events with background knowledge and media ,and linking events together may change the way people or systems exploit data.

This thesis thoroughly describes the different steps aiming to realize the vision of the Web of events having as a foundation the Social Web and harnessing the Semantic Web technologies. The work presented does not focus on the building process, but also on reusing the new representation of events in various areas including Web applications and personalization. The contributions made are:

- **Data Structuring**

  A milestone towards the Web of events is to semantically model what an "event" is. Due to its inherently multidimensional nature, we surveyed some different definitions, and we retained the one which represents the most described aspects. This definition is based on the *Ws* questions: *What*, *When*, *Where* and *Who*. To formalize the event definition, we opted for the LODE ontology as an interoperable model realized without any particular interpretation or perspective. This complies with our strategy to retrieve and model any type of event from the Social Web. On the other hand, the modeling of media was simply achieved by the reuse of popular ontologies in the domain.

- **Data Aggregation**

  Many Web directories contain event-centric data including calendar of events or captured media. Aggregating this data and exploring the explicit overlap between these directories are parts of the building process. Thus, we developed a framework that collects events and media, and exploits explicit metadata (e.g. machine tags, hashtag) to link them. We followed one design requirement which is the *flexibility*. The objective is to be able to flexibly add more event and media directories in the future.

- **Data Reconciliation**

  We particularly addressed two different tasks having in common the challenge of data heterogeneity. The former creates identity link between two same real-world instances and the latter aligns events with microposts at the sub-event level of granularity. For the first task, we surveyed existing automatic instance matching tools. Yet, none of them is able to overcome the heterogeneity found between event directories. As a solution, we proposed an approach based on the correlation and the coverage of predicates and taking into account various data types. As for the second task, we proposed a Named Entity-based approach to bridge the gap between the unstructured content of microposts and the structured description of events. The idea is to exploit the mapping between the classification of Named Entities and the concepts of the event ontology. Both tasks ensure a real-time reconciliation to face the dynamics of events.

- **Application and Analysis**

  We developed some Web Applications enabling new mechanisms to browse and search for events or to create events in a controllable way. Our experience highlighted some limitations to use RDF data within conventional Web technologies. We also showed the importance to design a simple data model at the expense of expressivity. Lastly, we explored the benefits of Linked Data to uncover behavioral aspects and to improve the user profiling.

- **Event Recommendation**

  We designed a hybrid recommender system in order to suggest personalized events. It is built on top of Semantic Web and combines content-based recommendation and collaborative filtering. It is shown that ontology-enabled feature extraction and enrichment with Linked Data significantly improve the performance. In addition, a user may be involved in many events, but interested in specific topics. Thus, we proposed a method to alleviate the impact of the topical diversity that may characterize a user profile. Results underlined the importance of the social information and the user interests modeling in event recommendation.

- **Community Detection in EBSN**

  We presented an approach to detect topical communities in event-based social network relying on structural and content features. The links between events and media were used to construct event-based networks from media directories. We also built networks using the co-attendance information obtained from event directories. The evaluation results shed light on the difference between these networks in terms of users' interactions.

In summary, the contributions achieved pave the way to build the Web of events as part of Linked Data. The main idea is to bring together event-centric data into a unified structured knowledge with the flexibility and depth afforded by the Semantic Web technologies. As rich data made available in the Linked Data cloud, one can expect efficient supports to browse, search and visualize rich data. The work presented in this thesis goes beyond this fact and further demonstrates the utility of the Semantic Web in other tasks such as personalization, user modeling or comparative analysis. Although focused on events, some proposed approaches could be easily propagated to other domains such as movie recommendation or community detection in social media.

8.2 Perspectives

A growing number of RDF datasets continuously feed the Linked Data cloud covering a multitude of diverse domains. This thesis specifically targeted the event domain to build and leverage a meaningful knowledge base. Still, it could be extended by the following future directions:

- **Enrichment**

  One enhancement is to enrich EventMedia dataset with other popular Web services such as Facebook and Eventbrite[1]. As such, we can increase the overlap in terms of coverage and benefit the assets of each website. Indeed, at the time of writing, the integration of Facebook was under development. Enrichment could also improve data modeling by incorporating useful vocabularies such as the Tickets ontology [71] or describing participants' friendships.

- **Connection of Events**

  Events sharing spatial-temporal context or having in common a specific topic or participants may have a connection between them. This reveals a key aspect in the Web of events which is to represent the natural relationships at different levels such as referential, structural and causal. While we only dealt with identity connection, the other types remain unexplored. This opens the door

[1] http://www.eventbrite.com

for future work exploring more meaningful connections. Moreover, the existing approaches mostly address a specific domain (e.g. historical [35]) and focus on specific event attributes (e.g. time [83]). There is a need for a formal specification that takes into account all the event attributes proving its efficiency to be applied in different domains (e.g. social, political).

- **Temporal Dynamics**

  Temporal dynamics is an important aspect that recently drives the way to design computing applications. The growth of online activities has led to new challenges about how to handle streaming data, instead of static files, which needs more efficiency and scalability. Moreover, data may shift over time, a fact that may impact many tasks such as reconciliation or recommendation. In instance matching, solving at the same time the high heterogeneity and temporal dynamics is a quite challenging problem. Our strategy focused on the heterogeneity problem still need a ground truth to learn the correlation and the coverage of properties. In order to face a future evolution, one solution that can be sought is to automatically generate a ground truth or to fully rely on an unsupervised method. Temporal dynamics has also an impact on event recommendation. Unlike a classic product, an event is ephemeral, and as such, the list of events in the profile of a very active user become unmanageable. To solve this, one simple approach commonly used is to discard irrelevant instances using a time window [173]. In our scenario, one can run SPARQL queries to simply index recent events, which raises the question about the effective window size and its impact on the system performance.

- **Scalable Recommender System**

  The information overload is a well-known problem that prevents users from easily making decision in an online service even when supporting browsing and searching capabilities. To overcome this problem in event-based service, we proposed a hybrid recommender system based on the classical Vector Space Model (VSM). Although efficient to provide personalized events, our approach has a serious drawback of scalability since the time complexity is linear to the number of events (i.e. documents in VSM). Considering this limitation, several optimization techniques found in the literature could be integrated to speed up the computation. One technique is to reduce complexity in VSM by pruning unnecessary similarity comparisons. This can be ensured by the high-dimensional similarity search techniques such as the popular indexing method named Locality Sensitive Hashing (LSH) [59]. Another solution worth to be investigated is the multi-relational learning using tensor factorization which can be applied in Linked Data. This is particularly the goal of Rescal-ALS, a scalable tool that represents entities in a latent space enabling efficient information propagation via the dependency structure [139].

- **Community-based Recommendation**

  Exploiting community detection for recommendation has been the subject of numerous research studies. It is also an indirect way to assess the quality of the identified communities. Indeed, it has been shown that taking advantage from a collective behavior of users is one solution to alleviate the cold-start problem [154, 16] or to diversify recommendation [53]. In this perspective, our recommender system could be improved by the integration of community detection approach applied on event-based social network. Another similar direction is to build a signed network from user interactions as has been proposed by Maniu et al. [122], which can be used to build a trust-aware recommender system.

Appendix A

List of Publications

A.1 Journals

- Khrouf, Houda; Troncy, Raphaël "De la modélisation sémantique des événements vers l'enrichissement et la recommandation", Revue d'Intelligence Artificielle, Numéro spécial Ingénierie des connaissances (under review).
- Khrouf, Houda; Milicic, Vuk; Troncy, Raphaël "Mining events connections on the social web: Real-time instance matching and data analysis in EventMedia", Web Semantics: Science, Services and Agents on the World Wide Web, 2014, http://dx.doi.org/10.1016/j.websem.2014.02.003.
- Khrouf, Houda; Troncy, Raphaël "EventMedia: a LOD Dataset of Events Illustrated with Media", Semantic Web Journal, Special Issue on Linked Dataset descriptions, IOS Press 2014.

A.2 Conferences and Workshops

- Khrouf, Houda; Troncy, Raphaël "Topical Community Detection in Event-based Social Network" (submitted to the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining), 24-27 August 2014, New York, USA.
- Khrouf, Houda; Troncy, Raphaël "Hybrid event recommendation using linked data and user diversity". In Proceedings of the 7th ACM Conference on Recommender systems (RecSys), 12-16 October 2013, Hong Kong, China. *(acceptance rate: 24%)*
- Buschbeck, Sven; Troncy, Raphaël; Jameson, Anthony; Khrouf, Houda; Spirescu, Adrian; Suominen, Osma; Schneeberger, Tanja; Hyvönen, Eero "Parallel faceted browsing". In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI), Interactivity Track, April 27-May 2, 2013, Paris, France.
- Khrouf, Houda; Milicic, Vuk; Troncy, Rapahël "EventMedia live: Exploring events connections in real-time to enhance content". In Semantic Web Challenge at the 11th International Semantic Web Conference (ISWC), November 11-15, 2012, Boston, USA. **1^{st} Prize Winner of the Semantic Web Challenge**.

- Buschbeck, Sven; Jameson, Anthony; Troncy, Raphaël; Khrouf, Houda; Suominen, Osma; Spirescu, Adrian "A demonstrator for parallel faceted browsing". In Intelligent Exploration of Semantic Data Workshop (IESD 2012), October 8-12, 2012, Galway, Ireland. **Winner of the IESD Challenge**.

- Khrouf, Houda; Atemezing, Ghislain; Rizzo, Giuseppe; Troncy, Raphaël; Steiner, Thomas "Aggregating social media for enhancing conference experiences". In Proceedings of the 1st International Workshop on Real-Time Analysis and Mining of Social Streams (RAMSS), June 4, 2012, Dublin, Ireland.

- Khrouf, Houda; Atemezing, Ghislain; Steiner, Thomas; Rizzo, Giuseppe; Troncy, Raphaël "Confomaton: A conference enhancer with social media from the cloud". In Proceedings of the 9th Extended Semantic Web Conference (ESWC), Demo Track, May 27-31, 2012, Heraklion, Crete.

- Khrouf, Houda; Troncy, Raphaël "EventMedia: visualizing events and associated media". In Proceedings of the 10th International Semantic Web Conference (ISWC), Demo Track, October 23-27, 2011, Bonn, Germany.

- Khrouf, Houda; Troncy, Raphaël "EventMedia Live: reconciliating events descriptions in the web of data". In Proceedings of the 6th International Workshop on Ontology Matching (OM), October 23-27, 2011, Bonn, Germany.

- Khrouf, Houda; Troncy, Raphaël "Réconcilier les événements dans le web de données". In Proceedings of the 22nd Journées Francophones d'Ingénierie des Connaissances (IC), May 16-20, 2011, Chambéry, France.

A.3 Archived Technical Reports

- Troncy, Raphaël; Khrouf, Houda; Shaw, Ryan; Hardman, Lynda "Specification of an event model for representing personal events". ALIAS Deliverable D4.1, 2011, http://deliverables.aal-europe.eu/call-2/alias/d4-1-specification-of-an-event-model-for-representing-personal-events.

- Troncy, Raphaël; Khrouf, Houda; Atemezing, Ghislain; Fialho, Andrsé; Hardman, Lynda "Module for knowledge enrichment of event descriptions". ALIAS Deliverable D4.3, 2011, http://deliverables.aal-europe.eu/call-2/alias/d4-3-module-for-knowledge-enrichment-of-event-descriptions.

- Khrouf, Houda; Troncy, Raphaël; Milicic, Vuk "Module for personalized discovery of new contacts on line". ALIAS Deliverable D4.4, 2013, http://deliverables.aal-europe.eu/call-2/alias/d4-4-module-for-personalized-discovery-of-news-contacts-on-line.

- Khrouf, Houda; Troncy, Raphaël "Module for retrieval of opinionated content". ALIAS Deliverable D4.5, 2013, http://deliverables.aal-europe.eu/call-2/alias/d4-5-module-for-retrieval-of-opinionated-content.
- Khrouf, Houda; Troncy, Raphaël "Module for topics recommendation". ALIAS Deliverable D4.7, 2013, http://deliverables.aal-europe.eu/call-2/alias/d4-7-module-for-topics-recommendation.
- Scharffe, Franois; Fan, Zhengjie; Ferrara, Alfio; Khrouf, Houda; Nikolov, Andriy "Methods for automated dataset interlinking". Datalift Deliverable D4.1, 2011, http://hal.inria.fr/hal-00793435.
- Euzenat, Jérôme; Abadie, Nathalie; Bucher, Bndicte; Fan, Zhengjie; Khrouf, Houda; Luger, Michael; Scharffe, Franois; Troncy, Raphaël "Dataset interlinking module". Datalift Deliverable D4.2, 2011, http://hal.inria.fr/hal-00793433.

Appendix B

Optimization Techniques

In this appendix, we overview some technical aspects used in this thesis. More precisely, we describe two artificial intelligence techniques namely the Genetic Algorithms and the Particle Swarm Optimization widely used in optimization problems.

B.1 Genetic Algorithms (GAs)

Genetic Algorithms are stochastic methods inspired by the mechanism of natural evolution and genetic inheritance [182]. GAs are are one of the most popular evolutionary algorithms widely used for solving optimization problems in many areas such as machine learning and image processing. The idea behind is that the best solution can be found by combining the "good" parts of other solutions.

In GAs, a population is a set of *chromosomes* (candidate solutions) and each chromosome denotes a set of *genes*. The content of each gene is called *allele*. A key component in GAs is the setting of a fitness criterion which accurately evaluates the quality of candidate solutions. First, a population of chromosomes are randomly generated and evaluated using the fitness function. The chromosomes having higher fitness values than others are stochastically selected, recombined and mutated to produce a new population for the next generation. To achieve this, GA has a set of key operators, namely *selection*, *crossover* and *mutation*. The selection operator is used to select chromosomes called *parents* to create the descendants of the next generation. The selection usually favored fitter patents, and there are approaches proposed in the literature. One example is the *Stochastic Universal Sampling (SUS)* developed by Baker [8], which is used in this thesis. Consider a line where each chromosome occupies a segment proportional to the chromosome's fitness. *SUS* uses N equally spaced pointers placed over the line, where N is the number of selections required.

Once parents for new population are chosen, genetic operators are applied such as crossover and mutation. Crossover refers to the recombination of parents to form a child. In particular, we used the scattered crossover which creates a random binary mask, then selecting the genes where the mask is 1 from one parent, and the rest from other parent. In order to force the algorithm exploring new areas in the search space, mutation is performed which alters at least one gene in a chromosome according to a predefined probability. Mutation rarely occurs in nature, which can justify the typical value 0.01 generally used as a mutation probability. Finally, the algorithm stops

iterating when the optimal solution is produced or a maximal number of iterations is reached.

B.2 Particle Swarm Optimization (PSO)

It is a population-based stochastic optimization technique inspired by the social behavior of bird flocking or fish schooling [89]. PSO is similar to evolutionary algorithms and it was introduced in 1995 by Kennedy and Eberhat. Compared with GA, PSO is easy to implement with few parameters to adjust, and each individual benefits from its history whereas no such mechanism exists in GA. PSO has been successfully applied to solving a wide range of optimization problems in different fields such as robotics, image, neural network, and information retrieval.

PSO simulates a group of birds searching for food in a bounded area, where the best position is the one containing the highest density of food. At the beginning, all the birds start searching for food randomly. Each bird knows two positions: its own position (i.e. history) found with the most of food and the best position from the whole swarm. The birds will be guided by these two positions in the search process until optimal convergence.

The PSO algorithm initializes a population of random solutions called *swarms* or *particles*, and searches for the optimal solution 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 particles in the swarm. This means that if a particle discovers a promising new solution, all the other particles will move closer to it, exploring the region more thoroughly in the search process. Each particle i in the swarm has the following attributes: a current position x_i, a current velocity v_i, and a personal best position p_i in the search space, and the global best position p_{gbest} among all the p_i. In each iteration, the velocity and the position of each particle is updated as following:

$$v_i(t+1) = w.v_i(t) + c_1 r_1(p_i - x_i(t)) + c_2 r_2(p_{gbest} - x_i(t))$$
$$x_i(t+1) = x_i(t) + v_i(t+1)$$

where c_1 is the acceleration coefficient for each particle to move to its personal best position, c_2 is the acceleration coefficient to move to the global best position, r_1 and r_2 are random numbers uniformly distributed within [0,1], and w is the inertia weight which controls the contribution of a particle's previous velocity to its current velocity. The velocity and acceleration are responsible for changing the position of the particle to explore the space of all possible solutions, instead of using existing

solutions to reproduce. The personal and the global best positions are the optima of a predefined fitness function, respectively in each iteration and for all past iterations. In this thesis, to adjust some PSO parameters, we followed the setting recommended by Eberhart and Shi [52].

Appendix C

String Similarity

There are different classes of string similarity functions. In this appendix, we focus on the main classes surveyed in the literature and we overview the most popular functions in each class. The similarity formulas described in this appendix compare two strings s and t which are associated with two token sets $S = s_1, s_2, ..., s_n$ and $T = t_1, t_2, ..., t_m$, respectively. For the computation, we used the Similarity Metric Library available online[1].

C.1 Token-based Functions

The first family of the string similarity is the token-based functions which consider a string as a set of tokens. Intuitively, tokens also called "bag of words" are substrings generated by a tokenization function (e.g. typically by a whitespace) applied on the original string. Making use of token-based functions is advantageous to overcome the word swaps. For example, the similarity between *Mahatma Gandhi* and *Gandhi Mahatma* will be maximal as both strings share the same tokens. However, the main drawback of such functions is to penalize approximate tokens having few spelling variations. That is, the comparison of *brother* and *brothers* will be zero.

One popular function is the Jaccard similarity [81] which is the ratio of the intersection size and the union size of two token sets:

$$Jaccard(S,T) = \frac{|S \cap T|}{|S \cup T|}$$

Q-gram is another function which splits a string into small overlapping (i.e. common characters) units of size q. To obtain such units with the first and last characters of a string, we introduce a padding character (e.g. #). For example, the 3-grams of *Gandhi* is the set *(##G,#Ga,Gan,and,ndh,dhi,hi#,i##)*. Then, Jaccard function is typically used based on these tokens to compute the similarity score.

The drawbacks of Jaccard is that it is very sensitive to spelling errors and it significantly penalizes the unmatched tokens. In contrast, q-gram is less sensitive to spelling errors or to unmatched tokens. This comparison is illustrated in Table C.1.

[1] http://sourceforge.net/projects/simmetrics

String s	String t	Jaccard	3-gram
Johnny Depp	Johny Dep	0	0.75
sir Johnny Depp	Mr Johnny Depp	0.5	0.78

Table C.1: Comparison between Jaccard and 3-gram

Cosine distance is another typical token-based function used in Information Retrieval for high dimensional data. Given two n-dimensional vectors X and Y containing the weights of tokens, the Cosine distance is defined as the cosine angle between these two vectors:

$$Cosine(X,Y) = \frac{|X \cdot Y|}{\|X\| \cdot \|Y\|} = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$$

In particular, the TF-IDF Cosine is commonly used where each token has a weight according to Term Frequency-Inverse Document Frequency scheme. This scheme is composed of two measures: term frequency (tf) and inverse document frequency (idf). The intuition behind the term frequency is that the more often a token occurs in a given string, the higher is its contribution to the similarity. In contrast, the inverse document frequency assigns higher weights to rare tokens in all the corpus (all the strings or documents). For each token s_i from the string s, the IF-IDF score is:

$$tf\text{-}idf_{i,s} = tf_{i,s} \cdot log\left(\frac{D}{D(t_i)}\right)$$

where $tf_{i,s}$ is the term frequency of s_i in the string s, D is the number of the strings in the corpus, $D(i)$ is the number of strings that contain the token s_i in the corpus. The computation of Cosine distance can be enhanced by hashing functions due to the high sparsity of most vectors. The advantage to use Cosine distance is to take into account the relative importance of different tokens in long strings and text documents.

C.2 Character-based Functions

The second family of string similarity is the character-based function also called edit-based similarity. Unlike the token-based functions, a string is considered as an ordered sequence of characters instead of a set of tokens. They allow different "edit operations" necessary to transform one string to another such as deletion, insertion, substitution, and transposition of characters. The use of these functions is mainly performed on short strings to overcome spelling errors. However, the performance of these functions drastically decreases when changing the order of the tokens.

One popular function is the Levenshtein distance [112] that allows three edit operations which are the deletion, insertion, and substitution. The score is equal to

the minimum number of operations required to transform s to t. For example, to transform *Maria* to *Mario*, we need the replace a by o, which gives a similarity score equal to 1. The normalized score is equal to 0.8. One drawback of Levenshtein is that it is not adapted for some variations such as abbreviations (e.g. *Gandhi Mahatma* and *Gandhi M*) or extra prefix (e.g. *Sir Gandhi* and *Gandhi*).

A similar metric is the Jaro distance [84] which allows character transpositions and based on the number and the order of common characters. Two characters are considered to be common if they are equal and if the distance between their positions i and j within the two strings does not exceed H, where $H = 0.5 \times min(|s|, |t|)$. Given a set of common characters σ, a transposition occurs if the i^{th} common character of s is different from the i^{th} common character of t. Let θ is half the number of transpositions, Jaro is computed as:

$$Jaro(s,t) = \frac{1}{3} \times \left(\frac{|\sigma|}{|s|} + \frac{|\sigma|}{|t|} + \frac{|\sigma| - \theta}{|\sigma|} \right)$$

Jaro distance performs well when there is few spelling variations. However, as common characters have to occur in a specific distance, variations such as a long prefix in one string (e.g. $s =$ *Doctor John Smith* and $t =$ *John Smith*) yields a low similarity of 0.46. A variant of Jaro distance, called Jaro-Winkler similarity [180] uses the length of the longest common prefix to emphasize matches in the first p characters of the two strings. For example the Jaro similarity between *John S* and *John Smith* is 0.86, while the Jaro-Winkler score is 0.94.

C.3 Hybrid Functions

To overcome the limitations of character and token based functions, the metrics in the third family combines both of them, also referred as hybrid functions.

Extended Jaccard similarity is a hybrid function proposed to also include not only the equals tokens, but also the similar ones in the the original Jaccard function [177, 6]. Consider *TokenSim* be a string similarity metric that compares two tokens s_i and t_j, and θ is the related threshold, the set of shared similar tokens between s and t is defined as:

$$Shared(s,t) = \{(s_i, t_j) | s_i \in S \wedge t_j \in T : TokenSim(s_i, t_j) \geqslant \theta\}$$

The set of unique or unmatched tokens in s is defined as:

$$Unique(s) = \{s_i | s_i \in S \wedge t_j \in T \wedge (s_i, t_j) \notin Shared\}$$

Similarly, we define the set $Unique(t)$ for the string t. This has been extended by a function that gives weights w to matched and unmatched tokens, which are combined using an aggregation function Ag. The hybrid Jaccard is defined as:

$$matched = Ag_{(s_i,t_j)\in Shared(s,t)}\ w(s_i,t_j)$$

$$unmatched = Ag_{(s_i)\in Unique(s)}\ w(s_i) + Ag_{(t_j)\in Unique(t)}\ w(t_j)$$

$$HybridJaccard(s,t) = \frac{matched}{matched + unmatched}$$

Note that different weights could be given for the tokens in $Shared(s,t)$, $Unique(s)$, and $Unique(t)$. For instance, let s= *Mindy Smith* and t=*Minndy Smith Festival*, the hybrid Jaccard generates the following sets:

$$Shared(s,t) = \{(Mindy, Minndy), (Smith, Smith)\}$$

$$Unique(s) = \varnothing$$

$$Unique(t) = \{Festival\}$$

Assuming that the weights of matched tokens is their normalized Levenshtein similarity, and the weights of unmatched tokens is equal to 1. If the aggregate function Ag simply sums the weights, the hybrid Jaccard is:

$$HybridJaccard(s,t) = \frac{0.83+1}{0.83+1+0+1} = 0.64$$

Note that the score remains low due to the influence of unmatched tokens. The Token-Wise metric proposed in Section **??** follows the same rationale, but gives more importance to similar tokens. Moreover, the weight of unmatched tokens takes into account the fact that the two token sets have different sizes. In this example, the weight for unmatched tokens is equal to $\frac{2}{3} = 0.66$. We obtain higher score than hybrid Jaccard when using Token-wise:

$$Token\text{-}Wise(s,t) = \frac{2\times(0.83+1)}{2\times(0.83+1)+0.66\times(0+1)} = 0.84$$

Another hybrid function is the Monge-Elkan similarity [133] that matches every token s_i from s with the token t_j in t having the maximum similarity using *TokenSim* metric. Monge-Elkan is defined as:

$$MongeElkan(s,t) = \frac{1}{|S|}\sum_{i=1}^{|S|}\max_{j=1}^{|T|} TokenSim(s_i,t_j)$$

Given the strings s = *Mindy Smith* and t = *Minndy Smith Festival*, and using Levenshtein as *TokenSim*, the Monge-Elkan score is:

$$MongeElkan(s,t) = \frac{0.83+1}{2} = 0.91$$

Monge-Elkan is sensitive to the size of the first string. For instance, if t is the first string which is of length 3, the Monge-Elkan score decreases to 0.61.

The last hybrid function is called SoftTFIDF [34] which extends the Cosine similarity, following the same rationale as hybrid Jaccard. Let $CLOSE(\theta, S, T)$ be the set of words $s_i \in S$ such that there is $t_j \in T$ where $TokenSim(s_i, t_j) > \theta$, and $maxsim(s_i, t_j) = max(\{TokenSim(s_i, t_j) | t_j \in T\})$. The SoftTFIDF is defined as:

$$SoftTFIDF(s,t) = \sum_{s_i \in CLOSE(\theta,S,T)} \left(\frac{tf\text{-}idf_{s_i}}{\|X\|} \cdot \frac{tf\text{-}idf_{t_j}}{\|Y\|} \times maxsim(s_i, t_j) \right)$$

where X and Y are the vector representations of s and t containing the $tf\text{-}idf$ scores of related tokens, respectively. Given the strings $s =$ *Mindy Smith* and $t =$*Minndy Smith Festival* and unit weights for all the tokens (no corpus considered), the SoftTFIDF gives:

$$SoftTFIDF(s,t) = \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{3}} \times 0.83 + \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{3}} \times 1 = 0.75$$

Appendix D

Recommender Systems

Broadly speaking, the recommender systems are based on two popular strategies: the content-based filtering and the collaborative filtering. In the following, we overview the basic concepts about those techniques.

D.1 Content-based Recommendation

The content-based systems exploit the attributes characterizing an item or a user. They analyze the content information collected explicitly or implicitly to construct a user or an item profile. The matching between both profiles can be quantified using a variety of similarity distances such as Cosine similarity, Pearson correlation and Latent Semantic Analysis [42]. This kind of matching is also applied to discover people sharing similar interests. It is closely related to detecting documents of similar content in information retrieval field. A known successful realization of content-based filtering is the Music Genome Project which is used for the Internet radio service Pandora.com. In this project, a trained music system ranks each song based on hundreds of distinct musical characteristics. These attributes or genes capture not only a song's musical identity but also many significant qualities which are relevant to understanding listeners' musical preferences [102]. Another interesting study proposed by Chen at al. [31] compare different recommender algorithms in the IBM's enterprise social networking service called "Beehive". The authors underline that the pure content matching is the most effective to recommend unknown friends and in general diverse items. However, the content-based recommendation has the drawback to not take into account the information in preference similarity across individuals.

D.2 Collaborative Filtering Recommendation

The second strategy of recommender system is based on collaborative filtering(CF), a technique that does not need an explicit content profiling and purely rely on past user behavior [72]. It has been widely applied in many well-known services such as Amazon, Faceboook, LinkedIn, MySpace and Last.fm. The basis is to analyze the relationships between users and inter-dependencies among items to identify new user-item associations. In other words, the system makes automatic predictions (filtering) about the user interests based on the preferences of like-minded and similar users

(collaborating). The intuition behind is that if a person A has the same preference as a person B on an item, A is more likely to have B's preference on another item, as illustrated in Figure D.1.

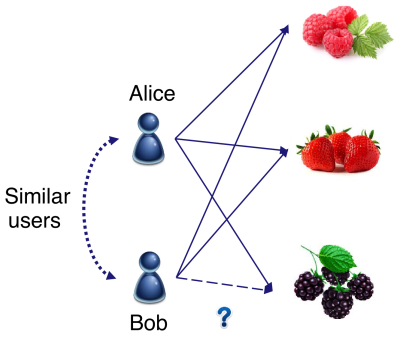

Figure D.1: user-based collaborative filtering: Alice has a crush on berry fruits, Bob also likes two of them. The recommender system understands that Alice and Bob have similar tastes, and Bob is recommended the Blackberry

There exists two primary categories of collaborative filtering which are memory-based and model-based approaches. The memory-based systems compute the similarity between users or, alternatively, between items based on users preferences data, thus detecting the neighbors of a given user or item. Indeed, the unknown rating value of the active user u for an item m is an aggregation of the ratings of users similar to u for the same item m, or an aggregation of the ratings of the user u to similar items of m. The model-based systems, on the other hand, use data mining and machine learning algorithms to estimate or learn a model from observed ratings to make predictions. A typical example is the latent factor model that discovers unobserved factors from ratings patterns. The underlying assumption is that there is a set of common hidden factors which explain a set of observations in co-occurrence data. More precisely, the similarity between users and items is simultaneously induced by some hidden lower-dimensional structure in the data. Recently, several matrix factorization methods [102] have been proposed as a successful realization of latent factor model. The users and items are simultaneously represented as unknown feature vectors within a user-item matrix. These feature vectors are learnt using low-rank approximations, so that they approximate the known preference ratings with respect to some loss measure. Despite the important success of collaborative filtering, it still suffers from three serious limitations: the sparsity problem where there are few ratings about items, the cold-start problem where items have no ratings, and the scalability where a large amount of users and items have to be analyzed.

Bibliography

[1] Z. Abedjan, T. Gruetze, A. Jentzsch, and F. Naumann. Profiling and mining rdf data with prolod++. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1198–1201, March 2014. 51, 67

[2] Maribel Acosta, Amrapali Zaveri, Elena Simperl, and Dimitris Kontokostas. Crowdsourcing Linked Data quality assessment. *ISWC 2013*, 2013. 43, 45, 53

[3] James Allan. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, 2002. 8

[4] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 1998. 7, 8, 13

[5] James F. Allen and George Ferguson. Actions and events in interval temporal logic. Technical report, University of Rochester, 1994. 13

[6] Rohit Ananthakrishna, Surajit Chaudhuri, and Venkatesh Ganti. Eliminating fuzzy duplicates in data warehouses. In *28th International Conference on Very Large Data Bases*, Hong Kong, China, 2002. 129

[7] Ahmad Assaf and Aline Senart. Data quality principles in the semantic web. In *Proceedings of the 2012 IEEE Sixth International Conference on Semantic Computing*, ICSC '12. 40, 42, 43

[8] James E. Baker. Reducing bias and inefficiency in the selection algorithm. In *2nd International Conference on Genetic Algorithms and Their Application*, Cambridge, Massachusetts, USA, 1987. 123

[9] H. Becker, M. Naaman, and L. Gravano. Event Identification in Social Media. In *12th International Workshop on the Web and Databases (WebDB'09)*, Providence, USA, 2009. 2

[10] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. In *5th International Conference on Weblogs and Social Media*, Barcelona, Spain, 2011. 13

[11] Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao. Prov-o: The prov ontology. Technical report, 2012. 46

[12] Tim Berners-Lee. Uniform Resource Identifier (URI): Generic Syntax - RFC 3986 (January 2005). http://tools.ietf.org/html/rfc3986. 15

[13] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. 14

[14] Diego Berrueta, Dan Brickley, Stefan Decker, and Sergio Fernández and-Christoph Görn et al. SIOC Core Ontology Specification (March 2010). http://rdfs.org/sioc/spec. 35

[15] Diego Berrueta, Sergio Fern√degndez, and Iv√degn Frade. Cooking http content negotiation with vapour. In *4th workshop on Scripting for the Semantic Web 2008 (SFSW2008). co-located with ESWC2008*, 2008. 51

[16] Aline Bessa, Alberto H. F. Laender, Adriano Veloso, and Nivio Ziviani. Alleviating the sparsity problem in recommender systems by exploring underlying user communities. In *6th Alberto Mendelzon International Workshop on Foundations of Data Management*, Ouro Preto, Brazil, 2012. 117

[17] Christian Bizer and Richard Cyganiak. Quality-driven information filtering using the wiqa policy framework. *Web Semant.* 39

[18] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009. 2, 18

[19] Christian Bizer, Jens Lehmann, Georgi Kobilarov, S√∂ren Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2009. 40, 53

[20] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. 85, 94, 103

[21] DCMI Usage Board. DCMI Metadata Terms (June 2012). http://dublincore.org/documents/dcmi-terms. 34

[22] Christophe Bohm, Felix Naumann, Ziawasch Abedjan, Fenz Dandy, Toni Grutze, Daniel Hefenbrock, Matthias Pohl, and David Sonnabend. ProÔ¨Åling Linked Open Data with ProLOD. *ICDE 2010*, 2010. 51

[23] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard. Web Services Architecture (February 2004). http://www.w3.org/TR/ws-arch. 25

[24] D Boyd and Kate Crawford. Six provocations for big data. *A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society*, 2011. 39, 65, 72

[25] Dan Brickley and R.V. Guha. RDF Schema 1.1 - W3C Recommendation (February 2014). http://www.w3.org/TR/rdf-schema. 16

[26] Dan Brickley and Libby Miller. FOAF Vocabulary Specification (January 2014). http://xmlns.com/foaf/spec. 34

[27] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *The seventh international conference on World Wide Web 7*, 1998. 55

[28] C Buil-Aranda and Aidan Hogan. SPARQL Web-Querying Infrastructure: Ready for Action? *International . . .*, 2013. 57

[29] Roberto Casati and Achille Varzi. Events. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2010. 7

[30] Soumen Chakrabarti, Byron E. Dom, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon Kleinberg. Mining the web's link structure, 1999. 55

[31] Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: recommending people on social networking sites. In *27th International Conference on Human Factors in Computing Systems*, Boston, MA, USA, 2009. 133

[32] Didier Cherix, Ricardo Usbeck, Andreas Both, and Jens Lehmann. CROCUS: Cluster-based ontology data cleansing. In *Proceedings of the 2nd International Workshop on Semantic Web Enterprise Adoption and Best Practice*, 2014. 52

[33] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004. 93, 94

[34] W. Cohen, P. Ravikumar, and S. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. In *1st International Workshop on Information Integration on the Web (IIWeb'03)*, pages 73–78, Acapulco, Mexico, 2003. 131

[35] Ilaria Corda, Vania Dimitrova, and Brandon Bennett. An ontological approach to unveiling connections between historical events. In *International Workshop on Intelligent Exploration of Semantic Data (IESD'12)*, Galway,Ireland, 2012. 13, 116

[36] Chris Cornelis, Xuetao Guo, Jie Lu, and Guanquang Zhang. A fuzzy relational approach to event recommendation. In *2nd Indian International Conference on Artificial Intelligence*, Pune, India, 2005. 3, 14, 91

[37] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *ACM Conference on Recommender Systems (RecSys'10)*, Barcelona, Spain, 2010. 89

[38] Juan David Cruz, Cécile Bothorel, and François Poulet. Entropy based community detection in augmented social networks. In *International Conference on Computational Aspects of Social Networks (CASoN)*, pages 163–168, Salamanca, Spain, 2011. 4, 94

[39] Richard Cyganiak and Anja Jentzsch. The Linking Open Data cloud diagram (September 2011). http://lod-cloud.net. 18

[40] Jeremy Debattista, Christoph Lange, and Sören Auer. daq, an ontology for dataset quality information. In *Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014.*, 2014. 54

[41] Jeremy Debattista, Santiago Londoño, Christoph Lange, and Sören Auer. LUZZU - A framework for linked data quality assessment. *CoRR*, abs/1412.3750, 2014. 54

[42] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. 133

[43] Renaud Delbru. Sindice at SemSearch 2010. *WWW10*, 2010. 56

[44] Renaud Delbru, Nickolai Toupikov, and Michele Catasta. Hierarchical link analysis for ranking web data. *The Semantic Web: Research and Applications*, 2010. 56, 68

[45] Jan Demter, Sören Auer, Michael Martin, and Jens Lehmann. Lodstats—an extensible framework for high-performance dataset analytics. In *EKAW 2012*, Lecture Notes in Computer Science (LNCS) 7603. Springer, 2012. 51

[46] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2001. 99

[47] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *8th International Conference on Semantic Systems, I-SEMANTICS*, Graz, Austria, 2012. 79, 80, 81, 92

[48] L Ding, Tim Finin, A Joshi, R Pan, and RS Cost. Swoogle: A semantic web search and metadata engine. *CIKM04*, 2004. 55, 68

[49] M. Doerr. The CIDOC Conceptual Reference Module: an Ontological Approach to Semantic Interoperability of Metadata. *AI Magazine*, 24(3):75–92, 2003. 34

[50] Xin Dong, Alon Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *ACM SIGMOD International Conference on Management of Data*, Baltimore, Maryland, USA, 2005. 3

[51] Simon Dooms, Toon De Pessemier, and Luc Martens. A user-centric evaluation of recommender algorithms for an event recommendation system. In *Workshop on Human Decision Making in RecSys'11*, Chicago, IL, USA, 2011. 91

[52] Russell C. Eberhart and Yuhui Shi. Particle swarm optimization: developments, applications and resources. In *IEEE Congress on Evolutionary Computation*, volume 1, pages 81–86, 2001. 125

[53] Maryam Fatemi and Laurissa Tokarchuk. A community based social recommender system for individuals & groups. In *5th International Conference on Social Computing*, Washington, DC, USA, 2013. 117

[54] A. Fialho, R. Troncy, L. Hardman, C. Saathoff, and A. Scherp. What's on this evening? Designing User Support for Event-based Annotation and Exploration of Media. In *1st International Workshop on EVENTS - Recognising and tracking events on the Web and in real life*, pages 40–54, Athens, Greece, 2010. 1, 11, 90

[55] A Flemming. Quality characteristics of linked data publishing datasources, 2010. 43, 44, 45, 46

[56] Giorgos Flouris, Yannis Roussakis, and M Poveda-Villalón. Using provenance for quality assessment and repair in linked open data. pages 1–12, 2012. 43, 44, 46, 56

[57] Conceptual Framework, Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, and Jens Lehmann. Quality Assessment Methodologies for Linked Open Data. *Under review, Semantic Web Journal*, 2012. 40, 43, 44, 45, 46, 47

[58] C Fürber and M Hepp. SWIQA - A Semantic Web information quality assessment framework. *ECIS 2011*, 2011. 53

[59] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *25th International Conference on Very Large Data Bases*, Edinburgh, UK, 1999. 100, 116

[60] G. Gouriten and P. Senellart. API BLENDER: A Uniform Interface to Social Platform APIs. In *21st World Wide Web Conference*, Lyon, France, 2012. 26

[61] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *National Academy of Sciences of the United States of America*, 101:5228–5235, 2004. 103

[62] W3C OWL Working Group. OWL 2 Web Ontology Language - W3C Recommendation (December 2012). http://www.w3.org/TR/owl2-overview. 17

[63] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993. 17

[64] Christophe Guéret, Paul T. Groth, Claus Stadler, and Jens Lehmann. Assessing linked data mappings using network measures. In *The 9th Extended Semantic Web Conference*, 2012. 43, 45, 56

[65] Junwei Han, Jianwei Niu, Alvin Chin, Wei Wang, Chao Tong, and Xia Wang. How online social network affects offline events: A case study on douban. In *9th International Conference on Ubiquitous Intelligence and Computing*, Fukuoka, September, 2012. 98

[66] Patricia Harpring. *Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works*. Getty Research Institute, 2010. 52

[67] Andreas Harth, Sheila Kinsella, and Stefan Decker. Using naming authority to rank data and ontologies for web search. *ISWC 2009*, 2009. 56

[68] Olaf Hartig and Jun Zhao. Using web data provenance for quality assessment. In *In: Proc. of the Workshop on Semantic Web and Provenance Management at ISWC*, 2009. 56

[69] Bernhard Haslhofer and Niko Popitsch. DSnotify: Detecting and fixing broken links in linked data sets. In *8th International Workshop on Web Semantics (WebS ’09), co-located with DEXA 2009*, 2009. 56

[70] Benjamin Heitmann, Richard Cyganiak, Conor Hayes, and Stefan Decker. An empirically grounded conceptual architecture for applications on the web of data. *Trans. Sys. Man Cyber Part C*, 42:51–60, 2012.

[71] Martin Hepp. Tickets Ontology. http://purl.org/tio/ns. 115

[72] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *the 2000 ACM Conference on Computer Supported Cooperative Work*, pages 241–250, Philadelphia, Pennsylvania, United States, 2000. 133

[73] Antonio Garrote Hernández and María N. Moreno García. Restful writable apis for the web of linked data using relational storage solutions. In *Workshop on Linked Data on the Web (LDOW'11)*, Hyderabad, India, 2011.

[74] Jerry R. Hobbs and Feng Pan. Time Ontology in OWL (September 2006). http://www.w3.org/TR/owl-time. 34

[75] Thomas Hofmann. Probabilistic latent semantic indexing. In *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, CA, USA, 1999. 94

[76] Aidan Hogan, Andreas Harth, and Stefan Decker. Reconrank: A scalable ranking method for semantic web data with context. In *In 2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006. 55

[77] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. Weaving the pedantic web. *LDOW 2010*, 2010. 43, 44, 45, 46, 47, 52

[78] Aidan Hogan, JüRgen Umbrich, Andreas Harth, Richard Cyganiak, Axel Polleres, and Stefan Decker. An empirical survey of linked data conformance. *Web Semant.*, 2012. 43, 44, 46

[79] Renato Iannella and James McKinney. vCard Ontology For describing People and Organisations (September 2013). http://www.w3.org/TR/vcard-rdf. 34

[80] Antoine Isaac and Ed Summers. Skos simple knowledge organization system primer. World Wide Web Consortium, Working Draft WD-skos-primer-20080829, August 2008. 44, 46

[81] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901. 127

[82] Ramesh Jain. Toward eventweb. *IEEE Distributed Systems Online*, 8, 2007. 13

[83] Vikramaditya R. Jakkula and Diane J. Cook. Learning temporal relations in smart home data. In *2nd International Conference on Technology and Aging*, Toronto, Canada, 2007. 116

[84] Matthew A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84:414–420, 1989. 129

[85] Joseph. M. Juran and A. Blanton Godfrey. *Juran's quality handbook*. McGraw Hill, 1999. 39

[86] Beverly K. Kahn, Diane M. Strong, and Richard Y. Wang. Information quality benchmarks: product and service performance. *Communications of the ACM*, 2002. 39

[87] Mehmet Kayaalp, Tansel Özyer, and Sibel Tariyan Özyer. A collaborative and content based event recommendation system integrated with data collection scrapers and services at a social networking site. In *International Conference on Advances in Social Networks Analysis and Mining*, Athens, Greece, 2009. 91

[88] C. Maria Keet, María del Carmen Suárez-Figueroa, and María Poveda-Villalón. The current landscape of pitfalls in ontologies. In *KEOD 2013 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Vilamoura, Algarve, Portugal, 19-22 September, 2013*, 2013. 43, 44, 45, 46, 57

[89] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995. 124

[90] Houda Khrouf, Ghislain Atemezing, Giuseppe Rizzo, Raphaël Troncy, and Thomas Steiner. Aggregating Social Media for Enhancing Conference Experience. In *1st International Workshop on Real-Time Analysis and Mining of Social Streams (RAMSS'12)*, Dublin, Ireland, 2012. 5, 96

[91] Houda Khrouf, Ghislain Atemezing, Thomas Steiner, Giuseppe Rizzo, and Raphaël Troncy. Confomaton: A conference enhancer with social media from the cloud. In *ESWC 2012, 9th Extended Semantic Web Conference*, Heraklion, Greece, 2012. 6

[92] Houda Khrouf, Vuk Milicic, and Rapahël Troncy. Eventmedia live: Exploring events connections in real-time to enhance content. In *Semantic Web Challenge at 11th International Semantic Web Conference*, Boston, USA, 2012. 6

[93] Houda Khrouf, Vuk Milicic, and Raphaël Troncy. Mining events connections on the social web: Real-time instance matching and data analysis in EventMedia. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 2014. 5

[94] Houda Khrouf and Raphaël Troncy. EventMedia : visualizing events and associated media. In *Demo Session at the 10th International Semantic Web Conference*, Bonn, Germany, 2011. 6

[95] Houda Khrouf and Raphaël Troncy. Eventmedia live: Reconciliating events descriptions in the web of data. In *6th International Workshop on Ontology Matching (OM'11)*, Bonn, Germany, 2011. 5

[96] Houda Khrouf and Raphaël Troncy. Eventmedia: a LOD dataset of events illustrated with media. *Semantic Web Journal, Special Issue on Linked Dataset descriptions*, 2012. 5, 37

[97] Houda Khrouf and Raphaël Troncy. Hybrid event recommendation using linked data and user diversity. In *7th ACM Conference on Recommender Systems*, Hong Kong, China, 2013. 6, 79

[98] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 1999. 55

[99] Joseph A. Konstan and John Riedl. Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22, 2012. 93

[100] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd International Conference on World Wide Web*, 2014. 54

[101] Dimitris Kontokostas, Amrapali Zaveri, S Auer, and J Lehmann. TripleCheckMate: A Tool for Crowdsourcing the Quality Assessment of Linked Data. *4th Conference on Knowledge Engineering and Semantic Web*, 2013. 53

[102] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer Society*, 42, 2009. 133, 134

[103] Carl Lagoze and Jane Hunter. The abc ontology and model. In *International Conference on Dublin Core and Metadata Applications 2001*, Tokyo, Japan, 2001. 34

[104] T. K Landauer and S. Dumais. Latent semantic analysis. *Scholarpedia*, 3(11):4356, 2008. 99

[105] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998. 80

[106] Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification - W3C Recommendation (February 1999). http://www.w3.org/TR/1999/REC-rdf-syntax-19990222. 15

[107] Page Lawrence, Brin Sergey, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998. 39, 55

[108] Danielle Hyunsook Lee. Pittcult: trust-based cultural event recommender. In *ACM Conference on Recommender Systems (RecSys'08)*, Lausanne, Switzerland, 2008. 91

[109] WonSuk Lee, Werner Bailer, Tobias Bürger, Pierre-Antoine Champin, and Jean-Pierre Evain et al. Ontology for Media Resources 1.0 (February 2012). http://www.w3.org/TR/mediaont-10. 35

[110] Jens Lehmann and Soeren Sonnenburg. Dl-learner: Learning concepts in description logics. *Journal of Machine Learning Research*, 2009. 52

[111] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Statistical properties of community structure in large social and information networks. In *17th International Conference on World Wide Web*, WWW '08, pages 695–704, New York, NY, USA, 2008. 107

[112] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707–710, 1966. 128

[113] Liyun Li and Nasir Memon. Mining groups of common interest: Discovering topical communities with network flows. In *9th International Conference on Machine Learning and Data Mining in Pattern Recognition*, Berlin, Heidelberg, 2013. 94

[114] Xiaoli Li, Aloysius Tan, Philip S. Yu, and See-Kiong Ng. Ecode: Event-based community detection from social networks. In *Database Systems for Advanced Applications*, Hong Kong, China, 2011. 93, 95, 98, 100

[115] Zhiwei Li, Bin Wang, Mingjing Li, and Wei-Ying Ma. A probabilistic model for retrospective news event detection. In *28th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 106–113, Salvador, Brazil, 2005. 7

[116] Guoqiong Liao, Yuchen Zhao, Sihong Xie, and Philip S. Yu. An effective latent networks fusion based model for event recommendation in offline ephemeral social networks. In *22nd ACM International Conference on Information and Knowledge Management*, San Francisco, USA, 2013. 14

[117] X. Liu, R. Troncy, and B. Huet. Finding media illustrating events. In *1st ACM International Conference on Multimedia Retrieval*, Trento, ITALIE, 2011. 2, 14

[118] Xingjie Liu, Qi He, Yuanyuan Tian, Wang-Chien Lee, John McPherson, and Jiawei Han. Event-based social networks: Linking the online and offline social worlds. In *18th ACM SIGKDD conference on Knowledge Discovery and Data Mining*, KDD'12, Beijing, China, 2012. 86, 93, 95, 96, 98

[119] Henry Living. Review of: Hedden, heather. the accidental taxonomist medford, nj: Information today, inc., 2010. *Inf. Res.*, 2010. 44, 45

[120] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: Efficient indexing for high-dimensional similarity search. In *33rd International Conference on Very Large Data Bases*, Vienna, Austria, 2007. 100

[121] Christian Mader, Bernhard Haslhofer, and Antoine Isaac. Finding quality issues in SKOS vocabularies. *Theory and Practice of Digital Libraries*, 2012. 43, 44, 45, 46, 47, 52

[122] Silviu Maniu, Bogdan Cautis, and Talel Abdessalem. Building a signed network from interactions in wikipedia. In *Databases and Social Networks*, Athens, Greece, 2011. 117

[123] Nicolas Marie, Fabien Gandon, Myriam Ribière, and Florentin Rodio. Discovery hub: On-the-fly linked data exploratory search. In *The 9th International Conference on Semantic Systems*, 2013. 40

[124] Michael Martin and Sören Auer. Categorisation of semantic web applications. In *4th International Conference on Advances in Semantic Processing*, Florence, Italy, 2010.

[125] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001. 99

[126] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *The 7th International Conference on Semantic Systems*, 2011. 40

[127] PN Mendes, Hannes Mühleisen, and Christian Bizer. Sieve: linked data quality assessment and fusion. *LWDM2012 - Proceedings of the 2012 Joint EDBT*, 2012. 53

[128] Peter Mika. *Social Networks and the Semantic Web*, volume 5 of *Semantic Web and Beyond*. Springer, 2007. 14

[129] A. Miles and S. Bechhofer. Skos simple knowledge organization system reference. w3C recommendation 18 August 2009., 2009. 44

[130] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Namespace Document (August 2009). http://www.w3.org/2009/08/skos-reference/skos.html. 35

[131] Einat Minkov, Ben Charrow, Jonathan Ledlie, Seth J. Teller, and Tommi Jaakkola. Collaborative future event recommendation. In *19th ACM Conference on Information and Knowledge Management*, Toronto, Ontario, Canada, 2010. 91

[132] Roberto Mirizzi, Tommaso Di Noia, Azzurra Ragone, Vito Claudio Ostuni, and Eugenio Di Sciascio. Movie recommendation with dbpedia. CEUR Workshop Proceedings, 2012. 40

[133] A. Monge and C. Elkan. The eld-matching problem: algorithm and applications. In *2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, Portland, Oregon, 1996. 130

[134] Óscar Muñoz-García and Raul Garcia-Castro. Guidelines for the specification and design of large-scale semantic applications. In *4th Annual Asian Semantic Web Conference*, Shanghai, China, 2009.

[135] Eetu Mkel. Aether – generating and viewing extended void statistical descriptions of rdf datasets. In *ESWC 2014 demo track, Springer-Verlag*, 2014. 51, 67

[136] Katsuko T. Nakahira, Masashi Matsui, and Yoshiki Mikami. The use of xml to express a historical knowledge base. In *16th International Conference on World Wide Web*, Banff, Alberta, Canada, 2007. 8

[137] Felix Naumann and Melanie Herschel. *An Introduction to Duplicate Detection.* Morgan and Claypool Publishers, 2010. 2

[138] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, 2004. 94, 105

[139] Maximilian Nickel and Volker Tresp. Tensor factorization for multi-relational learning. In *Machine Learning and Knowledge Discovery in Databases - European Conference*, Prague, Czech Republic, 2013. 116

[140] Georgios Paliouras. Discovery of web user communities and their role in personalization. *User Modeling and User-Adapted Interaction*, pages 151–175, 2012. 3, 93

[141] Toon De Pessemier, Sam Coppens, Kristof Geebelen, Chris Vleugels, Stijn Bannier, Erik Mannens, Kris Vanhecke, and Luc Martens. Collaborative recommendations with content-based filters for cultural activities via a scalable event distribution platform. *Multimedia Tools Appl.*, 58(1):167–213, 2012. 90

[142] Mara Poveda-Villaln, MariCarmen Surez-Figueroa, and Asuncin Gmez-Prez. Validating ontologies with OOPs! In *Knowledge Engineering and Knowledge Management.* Springer Berlin Heidelberg, 2012. 52

[143] Daniele Quercia, Neal Lathia, Francesco Calabrese, Giusy Di Lorenzo, and Jon Crowcroft. Recommending social events from mobile phone location data. In *10th IEEE International Conference on Data Mining*, Sydney, Australia, 2010. 84

[144] Y. Raimond, S. Abdallah, M. Sandler, and F. Giasson. The Music Ontology. In *8th International Conference on Music Information Retrieval (ISMIR'07)*, Vienna, Austria, 2007. 34

[145] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 103–110, Amsterdam, The Netherlands, 2007. 7

[146] Dave Reynolds, Jeni Tennison, and Leigh Dodds. Resource Description Framework (RDF) Model and Syntax Specification - W3C Recommendation (February 1999). https://code.google.com/p/linked-data-api/wiki/Specification.

[147] Giuseppe Rizzo, Thomas Steiner, Raphaël Troncy, Ruben Verborgh, José Luis Redondo García, and Rik Van de Walle. What fresh media are you looking for?: Retrieving media items from multiple social networks. In *International Workshop on Socially-aware Multimedia*, Nara, Japan, 2012. 26

[148] Ekkawut Rojsattarat and Nuanwan Soonthornphisaj. Hybrid Recommendation: Combining Content-Based Prediction and Collaborative Filtering. In *Intelligent Data Engineering and Automated Learning*, volume 2690, pages 337–344. Springer Berlin Heidelberg, 2003. 91

[149] Peter Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987. 109

[150] Matthew Rowe and Milan Stankovic. Aligning Tweets with Events: Automation via Semantics. *Semantic Web Journal*, 3(2):115–130, 2012. 3

[151] Edna Ruckhaus, Oriana Baldizan, and Maria-Esther Vidal. Analyzing linked data quality with liquate. In *OTM Workshops*, Lecture Notes in Computer Science, 2013. 54

[152] Anisa Rula and Amrapali Zaveri. Methodology for assessment of linked data quality. In *The 1st Workshop on Linked Data Quality co-located with 10th International Conference on Semantic Systems, LDQ@SEMANTiCS 2014, Leipzig, Germany, September 2nd, 2014.*, 2014. 42, 54

[153] Mrinmaya Sachan, Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. Using content and interactions for discovering communities in social networks. In *21st World Wide Web Conference*, Lyon, France, 2012. 95

[154] Shaghayegh Sahebi and William Cohen. Community-based recommendations: a solution to the cold start problem. In *Workshop on Recommender Systems*

and the Social Web, held in conjunction with ACM RecSys'11, Chicago, USA, 2011. 117

[155] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *19th International Conference on World Wide Web*, Raleigh, North Carolina, USA, 2010. 2, 13

[156] Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of The ACM*, 18:613–620, November 1975. 80

[157] Robert Scholes. Language, narrative, and anti-narrative. *Critical Inquiry*, 7(1):pp. 204–212, 1980. 7

[158] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. *IEEE Intelligent Systems*, 21(3), 2006. 33

[159] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating “word of mouth”. In *SIGCHI Conference on Human Factors in Computing Systems*, Denver, Colorado, USA, 1995. 93

[160] R. Shaw, R. Troncy, and L. Hardman. LODE: Linking Open Descriptions Of Events. In *4th Asian Semantic Web Conference (ASWC'09)*, pages 153–167, Shanghai, China, 2009. 8, 33, 34

[161] Evren Sirin, Michael Smith, and Evan Wallace. Opening, closing worlds - on integrity constraints. 2008. 52

[162] David A. Smith. Detecting and browsing events in unstructured text. In *25th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 73–80, Tampere, Finland, 2002. 7

[163] Dagobert Soergel. Thesauri and ontologies in digital libraries. In *JCDL*. ACM, 2005. 44, 47

[164] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas L. Griffiths. Probabilistic author-topic models for information discovery. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 306–315, 2004. 94

[165] Besiki Stvilia, Les Gasser, Michael B. Twidale, and Linda C. Smith. A framework for information quality assessment. *Journal of the American Society for Information Science and Technology*, 2007. 39

[166] Osma Suominen and Eero Hyvönen. Improving the quality of skos vocabularies with skosify. In *The 18th international conference on Knowledge Engineering and Knowledge Management*, 2012. 43, 44, 45, 46, 47, 52

[167] Osma Suominen and Christian Mader. Assessing and Improving the Quality of SKOS Vocabularies. *Journal on Data Semantics*, 2013. 43, 52

[168] Martin Szomszor, Terry R. Payne, and Luc Moreau. Using semantic web technology to automate data integration in grid and web service architectures. In *5th International Symposium on Cluster Computing and the Grid*, Cardiff, UK, 2005. 33

[169] Jiao Tao, Li Ding, and Deborah L. McGuinness. Instance data evaluation for semantic web-based knowledge management systems. In *HICSS*, pages 1–10. IEEE Computer Society, 2009. 52

[170] Berners-Lee Tim. Linked data. Technical report, W3C, 2006. 42, 51

[171] Nickolai Toupikov, J Umbrich, and Renaud Delbru. DING! Dataset ranking using formal descriptions. *WWW09*, 2009. 55, 56

[172] Raphaël Troncy, André T. S. Fialho, Lynda Hardman, and Carsten Saathoff. Experiencing events through user-generated media. In *1st International Workshop on Consuming Linked Data*, Shanghai, China, 2010. 1

[173] Alexey Tsymbal. The problem of concept drift: Definitions and related work. Technical Report TCD-CS-2004-15, The University of Dublin, Trinity College, Ireland, 2004. 116

[174] Mateja Verlic. Lodgrefine - lod-enabled google refine in action. In *I-SEMANTICS (Posters & Demos)*. CEUR-WS.org, 2012. 55

[175] RY Wang and DM Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 1996. 39

[176] Zhu Wang, Xingshe Zhou, Daqing Zhang, Dingqi Yang, and Zhiyong Yu. Cross-domain community detection in heterogeneous social networks. *Personal and Ubiquitous Computing*, 18(2):369–383, 2014. 95, 105

[177] Melanie Weis and Felix Naumann. Dogmatix tracks down duplicates in xml. In *ACM SIGMOD International Conference on Management of Data*, Baltimore, Maryland, 2005. 129

[178] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *5th International Conference on Weblogs and Social Media*, Barcelona, Spain, 2011. 13

[179] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graph. In *SIAM International Conference on Data Mining*, pages 274–285, Newport Beach, CA, USA, 2005. 94

[180] William E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau, 1999. 129

[181] Hao Wu, Vikram Sorathia, and Viktor Prasanna. When diversity meets speciality: Friend recommendation in online social networks. *ASE Human Journal*, 1:52–60, 2012. 85

[182] Jen yuan Yeh, Jung yi Lin, Hao ren Ke, and Wei pang Yang. Learning to rank for information retrieval using genetic programming. In *SIGIR Workshop on Learning to rank for Information Retrieval*, 2007. 123

[183] J.M Zacks, T.S Braver, M.A. Sheridan, D.I Donaldson, A.Z Snyder, J.M. Ollinger, RL Buckner, and M.E Raichle. Human brain activity time-locked to perceptual event boundaries. *Nature Neuroscience*, 4:2126–2431, 2001. 1

[184] Kuo Zhang, Juan Zi, and Li Gang Wu. New event detection based on indexing-tree and named entity. In *30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 215–222, Amsterdam, The Netherlands, 2007. 7

[185] Zhongying Zhao, Shengzhong Feng, Qiang Wang, Joshua Zhexue Huang, Graham J. Williams, and Jianping Fan. Topic oriented community detection through social objects and link analysis in social networks. *Knowledge-Based Systems*, 26:164–173, 2012. 4, 94, 95, 98, 101, 104

[186] Ding Zhou, Eren Manavoglu, Jia Li, C. Lee Giles, and Hongyuan Zha. Probabilistic models for discovering e-communities. In *15th International Conference on World Wide Web*, pages 173–182, New York, NY, USA, 2006. 94