UNIVERSITY OF MILANO - BICOCCA

DEPARTMENT OF COMPUTER SCIENCE, SYSTEMS AND COMMUNICATIONS

DOCTORAL SCHOOL IN COMPUTER SCIENCE - XXVI CYCLE

Ph.D. Thesis

Time-Related Quality Dimensions in Linked Data

Author: Anisa Rula

Advisors: Dr. Ing. Andrea Maurino
Dr. Matteo Palmonari
Tutor: Prof. Carlo Batini

Abstract

Over the last few years, there has been an increasing diffusion of Linked Data as a standard way to publish interlinked structured data on the Web, which allows users, and public and private organizations to fully exploit a large amount of data from several domains that were not available in the past. Although gathering and publishing such massive amount of structured data is certainly a step in the right direction, *quality* still poses a significant obstacle to the uptake of data consumption applications at large-scale. A crucial aspect of quality regards the dynamic nature of Linked Data where information can change rapidly and fail to reflect changes in the real world, thus becoming out-date.

Quality is characterised by different dimensions that capture several aspects of quality such as accuracy, currency, consistency or completeness. In particular, the aspects of Linked Data dynamicity are captured by Time-Related Quality Dimensions such as data currency. The assessment of Time-Related Quality Dimensions, which is the task of measuring the quality, is based on temporal information whose collection poses several challenges regarding their availability, representation and diversity in Linked Data. The assessment of Time-Related Quality Dimensions supports data consumers in their decisions whether information are valid or not.

The main goal of this thesis is to develop techniques for assessing Time-Related Quality Dimensions in Linked Data, which must overcome several challenges posed by Linked Data such as third-party applications, variety of data, high volume of data or velocity of data. The major contributions of this thesis can be summarized as follows: it presents a general settings of definitions for quality dimensions and measures adopted in Linked Data; it provides a large-scale analysis of approaches for representing temporal information in Linked Data; it provides a sharable and interoperable conceptual model which integrates vocabularies used to represent temporal information required for the assessment of Time-Related Quality Dimensions; it proposes two domain-independent techniques to assess data currency that work with incomplete or inaccurate temporal information and finally it provides an approach that enrich information with time intervals representing their temporal validity.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Linked Data is a paradigm that refers to a Web-scale knowledge base consisting of interlinked structured data published on the Web [22]. The increasing diffusion of Linked Data as a standard way to share knowledge in the Web allows users, and public and private organizations to fully exploit structured data from very large data sets that were not available in the past. Over the last few years, Linked Data developed into a large number of data sets with an open access from several domains [7] leading to the Linking Open Data cloud[1]. Although gathering and publishing such massive amount of structured data is certainly a step in the right direction for data consumption applications, data is as useful as its quality [11]. Similar to data in information systems and databases, Linked Data suffers from quality problems such as inconsistency, inaccuracy, out-of-dateness or incompleteness, which are frequent [72] and imply serious limitations to the full exploitation of such data.

Data quality is commonly conceived as *fitness for use* [86, 82, 141], which implies that data useful for a certain application or use case may not be useful for another application or use case [132]. To guarantee the full exploitation of the published or consumed Linked Data, it is important to evaluate Linked Data quality to understand whether it is appropriate for the task at hand before using it. For example, consider a consumer who executes a SPARQL query over the DBpedia data set to learn about cities that she is going to visit during her trip. In this case, it is rather neglectable if in relatively few cases the information is not complete. But, this data quality issue becomes crucial when it comes to taking far-reaching

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

decisions for developing an eTourism application. In this case, the quality of DBpedia is probably insufficient, as shown in [146].

Data quality is a multidimensional concept and may be measured along many dimensions such as accuracy, completeness, timeliness or trustworthiness [141]. Dimensions refer to abstract definitions and thus the assessment of data quality is based on measures that have to be defined to evaluate data quality. The process of measuring data quality is supported by metadata as well as data itself. Metadata plays an important role in supporting metrics evaluation since they store aspects of data relevant to data quality [11]. As an example, a user may measure how current is a data set based on the last modification date (e.g. the metadata provided by the `last modified` attribute). There exist many methodologies proposed in the information system and database community to identify and manage the quality of the published data, all addressing different aspects of quality assessment by proposing appropriate dimensions, measures and tools [12, 92, 118, 140]. Given the success of current methodologies for assessing and improving the quality of data in information systems and databases, one might question whether the methodologies of quality assessment in information systems and databases are appropriate to Linked Data.

The topic of Linked Data quality assessment has not yet received sufficient attention from the Linked Data community [147] and it poses a number of unique challenges. First, Linked Data refers to a Web-scale knowledge base consisting of interlinked published data from a multitude of *autonomous* information providers (variety of data). The quality of provided information may depend on the intention of the information provider. Second, the increasing diffusion of the Linked Data paradigm as a standard way to share knowledge on the Web allows consumers to fully exploit *vast amount of data* that were not available in the past (high volume of data). We are likely to find more low quality in Linked Data than in smaller data sets because in large data sets data are produced with automatic information processes which are often error prone. Third, data sets in Linked Data formats may often be used by *third-party applications* in ways not expected by the original creators of the data set. Fourth, Linked Data provides *data integration* through interlinking of data between heterogeneous data sources. The quality of integrated data will depend on the quality of original data sources. Last but not least relevant, Linked Data can be considered as a dynamic environment where information can change rapidly and cannot be assumed to be static (velocity of data) [83]. Changes in Linked Data sources should reflect changes in the real world [134, 35], otherwise data can soon become out-dated. Out-of-date information can reflect data inaccuracy problems and can deliver invalid information. For example, more up-to-date information should Data be preferred over less up-to-date information in data integration and fusion applications [104, 114].

It is important to identify adequate techniques for measuring quality of Linked Data. Despite the quality in Linked Data being an essential concept, few efforts are currently in place to provide quality assessment techniques or tools. Linked Data poses new challenges that are not handled before in other research areas.

The goal of this thesis is to provide a study on quality dimensions adopted to the Linked Data context and since the area of research on quality assessment is very broad, this thesis mainly focus on Time-Related Quality Dimensions due to the dynamic nature of Linked Data. Time-Related Quality Dimensions also referred to as data freshness are considered as the most important aspects of data quality for data consumers [127] and they play also an important role in the success of the information systems [141, 99].

1.1 Problem Statement

The information providers need to be aware about and deal with the dynamic nature of Linked Data [130], before integrating and presenting them to the users. Therefore, the assessment of Time-Related Quality Dimensions is crucial for information providers. On the other hand, the assessment of Time-Related Quality Dimensions is also important for users and applications that consume dynamic data since their assessment may provide support about the validity of information [97].

The assessment of Time-Related Quality Dimensions is based on metadata, that are, temporal meta-information, that is a specific type of temporal information. Temporal meta-information are particularly relevant because they associate facts and documents with versioning metadata and temporal validity. Examples of versioning metadata can be the update or the creation time of RDF data elements[2]. Examples of temporal validity can be a time interval indicating the validity of the fact.

[2] By RDF data elements we mean RDF triples, RDF documents or RDF graphs [67].

To acquire and use temporal meta-information we need to deal with the availability, representation and diversity of temporal meta-information in Linked Data which represent several challenges:

Availability of temporal meta-information - Time-Related Quality Dimensions assessment need available temporal meta-information. Some data sets may provide explicit and useful temporal meta-information such as the last update time or the update frequency, but other data sets may provide either incomplete and inaccurate, or implicit temporal meta-information. Thus, although in theory, Time-Related Quality Dimensions can be assessed based on existing methods, which rely on explicit temporal meta-information, in practice, these methods cannot be applied because temporal meta-information are not always available. Quality assessment techniques in Linked Data should also operate with implicit temporal meta-information;

Representation of temporal information - Time introduces a further dimension to data which cannot be easily represented in the data models currently used in Linked Data. Therefore, Time-Related Quality Dimensions assessment need to understand how temporal information are represented in Linked Data. Different approaches are adopted for representing temporal information as an additional component to annotate documents or facts. Therefore, there is a need to make consumers or applications aware about such representation approaches since the acquisition and understanding of such data is fundamental for the development of applications able to deal with dynamic data;

Diversity of temporal meta-information - Linked Data publishers may use different but semantically equivalent terminology to describe temporal meta-information across data sets, for example, data sets will use different vocabularies for expressing the last update time - which poses the user or applications with different terminology for stating the same information. This becomes a significant obstacle for applications consuming heterogeneous data sets and to overcome such problem, an alignment and mapping of heterogeneous temporal meta-information is needed in order to ensure a shared and interoperable conceptual model.

As such, temporal meta-information are used as basis to assess Time-Related Quality Dimensions across heterogeneous domains. The question of Time-Related

Quality Dimensions assessment would be trivial if all documents and facts carried correct and complete versioning metadata (associated usually with documents) and temporal validity intervals (usually associated with facts). In practice, however, one finds that versioning metadata and temporal validity are unreliable or unavailable. This becomes a significant obstacle for the assessment task.

Research Question

The goal of this thesis is to develop techniques, which enable consumers to measure Time-Related Quality Dimensions and improve the availability of temporal meta-information in Linked Data. We capture this goal in the following main research question that is substantiated in the thesis.

Given a heterogeneous Linked Data corpus, measures of Time-Related Quality Dimensions can be provided in a domain independent, and scalable manner over arbitrary Linked Data sources through a) techniques that acquire available temporal meta-information with heterogeneous representations b) propagation of temporal meta-information.

1.2 Main Contributions

The investigation of the outlined research question leads to the following contributions of this thesis, which also constitute the scientific accomplishment of the author.

Contribution I: ***Definitions of Linked Data quality dimensions.*** There has been of course recognition of the quality problems within the Linked Data community. However, Linked Data quality research field is currently evolving and cannot be considered mature enough to cover all the quality issues. Therefore, we examine the problem at its root, addressing the task of unifying, formalizing and adapting the definition for each data quality dimension from existing approaches in information systems and databases, web information systems and Semantic Web. We present the set of dimensions adopted in the Linked Data context, with a classification of quality dimensions. This study provides a broader context of quality dimensions and metrics in Linked Data which can serve as a starting point for researchers, data consumers and those implementing data quality protocols specifically for Linked Data (Chapter 4).

Contribution II: ***Analysis of approaches for temporal meta-information representation.*** We provide a large-scale analysis of approaches to represent temporal meta-information, including the principles that underlie such approaches. Temporal meta-information annotate documents or facts; these two levels of granularity allows us to assess Time-Related Quality Dimensions at document and fact level (Chapter 5).

Contribution III: ***Conceptual model of versioning metadata.*** We provide a sharable and interoperable conceptual model which aggregates those temporal meta-information used for the assessment of Time-Related Quality Dimensions, according to an alignment and mapping of heterogeneous temporal meta-information across different data sets (Chapter 6). This model comprises a broader context of temporal meta-information enabling the techniques of assessment to be domain independent.

Contribution IV: ***Assessment techniques for Time-Related Quality Dimensions.*** Facts and structured documents are annotated with unreliable or incomplete temporal meta-information; we propose two different techniques to overcome such issue (Chapter 6). The first measure based on the age of data elements propagates freshness of facts based on the freshness of the documents. The second measure is based on the delay between the Linked Data source and the original source where the original source has a versioning mechanism and each version is identified by a time point.

Contribution V: ***Improvement of temporal validity.*** In order to overcome the problem of incomplete or missing temporal meta-information associated with facts, we provided an approach for mapping facts to their temporal validity intervals. We devise a three-step approach that acquires evidence from both the Web of documents and the Web of Data (Chapter 7).

Research Method

Inspired by the industrial management concept called *T-shaped management* [61], we initially study all quality dimensions that are related to quality assessment in Linked Data (the horizontal part of the "T"), while the main goal of this work is to provide techniques for measuring Time-Related Quality Dimensions (the vertical "T"), in order to generate an understanding about the freshness in an open Web setting. In this way, we focus only on those dimensions that deals with the dynamic nature of the Web of Data, since these dimensions are crucial for the quality of Linked Data applications where it is of high relevance to keep huge amount of data up-to-date.

1.3 Thesis Outline

The remainder of the thesis is structured as follows:

- Chapter 2 introduces some core concepts and notation related to Semantic Web standards, Linked Data publishing and information quality which are used throughout the rest of the thesis.
- Chapter 3 positions this thesis with respect to related works. It gives an overview of the state-of-the-art techniques in Time-Related Quality Dimensions assessment in information systems and databases, and in Linked Data, at the intersection of which our work resides.
- Chapter 4 describes the study on quality dimensions and metrics in Linked Data. It provides definitions about quality dimensions and list metrics for assessing quality in Linked Data.
- Chapter 5 describes the approaches proposed in the literature for the representation of temporal information and discuss their adoption in well-known data sets. A large-scale experiment is provided in order to quantitatively investigate the adoption of these patterns in the LOD cloud.
- Chapter 6 introduces the metrics for measuring freshness of the information at document level. It further devise an approach of measuring fact freshness starting from the freshness of the documents. A conceptual model that

aggregates those temporal information used for the freshness assessment is provided, according to a large-scale analysis of the data sets in the LOD cloud. This model comprises a broader context of temporal meta-information at various data sets.

- Chapter 7 deals with an hybrid approach which comprises evidence from the Web of Data and the Web of documents for mapping facts to sets of time intervals.
- Chapter 8 finally summarizes the results and compares them to the motivation presented in this chapter. We collect and comment on research questions that remain open, and outline the expected future work and impact of the research topic.

Chapter 2

Foundations and Technical Background

This chapter introduces the foundations and the technical background for the work presented in this thesis. We first present in Section 2.1 some core Semantic Web standards, followed by Linked Data principles. Section 2.2 focuses on another background needed for our work, namely Temporal RDF. Finally, we introduce core concepts of information quality assessment in Section 2.3.

2.1 Web, Semantic Web and Linked Data

In order to understand the concept and value of Linked Data, it is important to first consider the technologies used by the "current" World Wide Web (Section 2.1.1), followed by Semantic Web standards (Section 2.1.2). Finally, we present the principles of Linked Data build upon the above technologies and standards (Section 2.1.3).

2.1.1 World Wide Web and the Rationale for Linked Data

The World Wide Web is considered as a global information space where Web documents are interlinked to other related Web documents through hyperlinks which allow users to browse between related documents. Web documents rely on a set of simple standards, such as Uniform Resource Identifiers (URIs) or HyperText

Markup Language (HTML). A URI identifies globally a Web document [16] and not only; a URI, as a globally unique identification mechanism, is used to identify also other Web resources such as real world objects (e.g., places, people or images). The Web makes use of URIs to enable interaction with other documents through specific protocols such as the application level protocol - Hypertext Transfer Protocol (HTTP) [42]. The content of the Web documents that all computers may understand is represented by the Hypertext Markup Language (HTML) which contains formatted natural language, digital images (e.g. JPEG), and other rendering instructions [121].

Despite the benefits the Web provides, most of the Web's content is designed for humans to read. Machines are not able to understand information for their convenient consumption. Consider for instance, one is interested to answer to the following question: "Which is the world capital with the highest population in the world?". We can get an answer for that question from the actual Web if either someone has performed and published the result of the query or if there exist a website having the data for download in a structured format which can be processed offline. Machines can parse Web pages for layout or routine processing, but in general, they cannot process the above query since no machine-readable structured data and semantics are made available by the respective sources such that they can subsequently be processed by machines. In general, data published on the Web are as raw dumps in formats such as CSV or XML, or marked up as HTML tables, sacrificing much of its structure and semantics. Further, Web documents as mentioned before, connect to each other through hyperlinks that are not semantically processable by machines.

Linked Data, which applies to the general architecture of the World Wide Web [80], engages Semantic Web standards such as RDF, RDFS, OWL and SPARQL to provide structured data such that they can be subsequently processed by machines. In the following section we give a short overview of the technologies provided from the Semantic Web respectively.

2.1.2 Semantic Web Standards

Semantic Web is conceived as an extension of the current Web, which enables sharing and re-use of data over the Web. Traditionally, the Semantic Web is represented as a "Semantic Web Layer Cake" [18] where each layer represents a

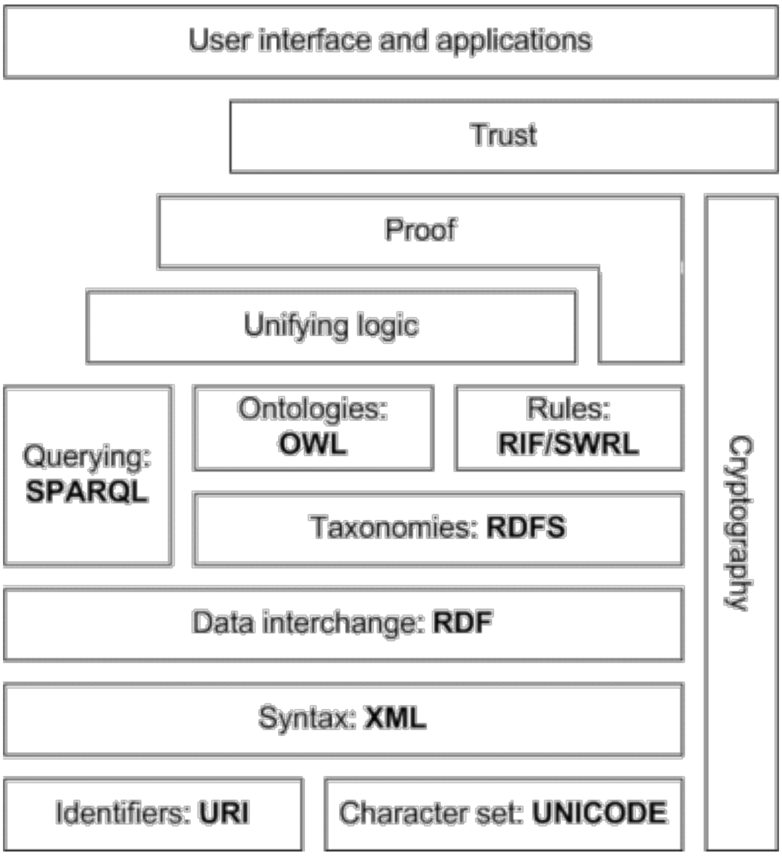


FIGURE 2.1: Semantic Web Layer Cake. Reprinted from http://w3.org/DesignIssues/diagrams/sweb-stack/2006a.png#sthash.M4s7TQTy.dpuf. Copyright (c) 2006 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All rights reserved.

technical part needed for its construction (see Figure 2.1). Here we shortly discuss some of the fundamental layers of the "cake": RDF, RDFS, OWL and SPARQL.

Resource Description Framework

In order to enable applications to process data on the Web, it is important to represent content expressed in a standard format. For this reason, a data model known as Resource Description Framework (RDF) [85] is used. The RDF enables machines to exchange structured data which are represented in a common data model. Information expressed in RDF are resources that can be exchanged between applications without loss of meaning. RDF is part of the W3C Recommendation [90].

The basic idea of the RDF data model is a statement represented by a triple containing three *RDF terms* in the form of subject-predicate-object. An RDF term is an element that can be of three types: URI, literal and blank node. There are two types of literals in an RDF data model: *plain* and *typed* literals. As URIs were discussed before, we now explain literals and blank nodes. Both plain and typed literals represent a literal value, like a string, number, or date. The plain literal is a string associated with a language tag [5]. A language tag (e.g. `"Lisbon"@en`) indicates a language such as English or Italian. The typed literal is a string associated with a datatype URI. A datatype URI is defined by the XML schema and indicates dates, integers and floating point numbers, e.g. `"1985-02-05"^^xsd:date`. A blank node is neither a URI nor a literal, but it just denotes the existence of some resources without a name. Blank nodes are used inside a document that contains an RDF description and cannot be referenced outside of their originating scope. Formally, a triple is defined as follows:

Definition 2.1 (RDF triple)**.** Given an infinite set $\mathcal{U}$ of URIs, an infinite set $\mathcal{B}$ of blank nodes, and an infinite set $\mathcal{L}$ of literals, a triple $\langle s, p, o \rangle \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$ is called an *RDF triple* where s, p, o represents the subject, predicate and object respectively of the triple.

Further, as the use of blank nodes is discouraged for Linked Data since blank nodes do not have consistent naming [22], in the following will be assumed that the subject and the predicate are URIs, while the object (also known as the property value) can be either a URI or a literal. Based on the type of the object, RDF triples can be distinguished in two types [67]:

- *Literal triples* is an RDF triple where the object is of type literal.
- *RDF links* is an RDF triple where the object is of type URI. Further, RDF links can be distinguished in: internal and external. While in the former case the link is provided between two resources belonging to the same data set, in the latter case the link is provided between two resources belonging to two different data sets.

An RDF data model can represent information as a directed graph that consists of nodes and edges where nodes refer to subjects or objects and edges refers to predicates which provide links between nodes. The datatypes nodes are represented

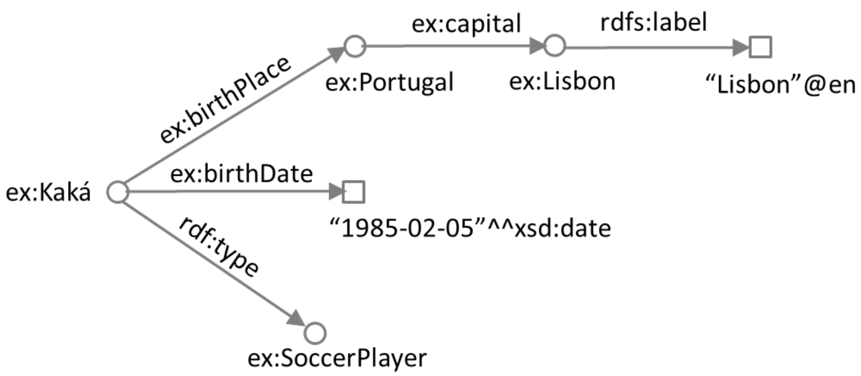


FIGURE 2.2: Example of RDF.

by squares. The term RDF graph has been adopted from the W3C Data Access Working Group [14, 66, 27] and formally is defined as follows:

Definition 2.2 (RDF graph)**.** An *RDF graph* $\mathcal{G} \subset (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$ is a finite set of RDF triples.

Example 2.1. *Figure 2.2 represents an example of an RDF graph containing five triples. The semantic of this graph is that the soccer player Kaka' was born on 5 February 1985 in Portugal and that Lisbon is the capital of Portugal. Within the graph, Kaka', Portugal, Lisbon and soccer player are identified by URI references. Date of birth and the label Lisbon are identified as labels and are represented by square nodes.*

RDF data can be stored by using the N-quad format; a quad is a quadruple $\langle s, p, o, c \rangle$ where c defines the context of an RDF triple $\langle s, p, o \rangle$; the context describes the provenance of a triple, often represented by - but not limited to - a *named graph*. The basic idea of the named graph is to introduce a graph naming mechanism, which takes a set of RDF triples and name this graph with a URI reference. Formally a named graph can be defined as follows:

Definition 2.3 (Named graph)**.** A *named graph* denoted as u_G, is an RDF graph G identified by a URI u ($u \in \mathcal{U}$).

Syntax for RDF

RDF needs a syntax in order to publish and interchange RDF data between information systems on the Web. Different serializations are proposed: RDF/XML [14], N-Triples [55], N3 [17], and Terse RDF Triple Language (Turtle) [1]. Throughout this thesis, we will use Turtle to represent RDF triples and graphs. Turtle defines a textual syntax and allows a compact and natural text form, with abbreviations for common usage patterns and datatypes. URIs are enclosed with brackets and may be abbreviated when URIs are repeated by using the symbol `@prefix` and a qualified name to be used in the document. Literals are represented between double quotes and may be given either a language suffix or a datatype URI by using the symbol `@` followed by the language tag and the symbol `^^` followed by any legal URI respectively. Blank nodes are represented by using the underscore prefix.

The RDF graph given in the Example 2.1 is written in the Turtle syntax as shown in Listing 2.1

Listing 2.1: Example of a set of RDF triples.

```
@prefix : <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix rdf: <http://www.w3.org/1999/02/22−rdf−syntax−ns#> .

ex:Kaka        rdf:type        ex:SoccerPlayer .
ex:Kaka        ex:birthDate    ”1985−02−05”^^xsd:date .
ex:Kaka        ex:birthPlace   ex:Portugal .
ex:Portugal    ex:capital      ex:Lisbon .
ex:Lisbon      rdfs:label      ”Lisbon”@en .
```

Semantic for RDF

RDF Schema. RDF is not specialized to name and describe classes of things and their relationships. That is the role of RDF vocabulary description language, RDF Schema (RDFS), used to define resources [27]. RDFS allows resources to be classified explicitly as classes or properties. All resources of a class share the same characteristics determined by the class. Resources can be instances of multiple classes and classes can have multiple instances. The most popular term is `rdf:type` (for brevity we use the "rdf" prefix for <`http://www.w3.org/1999/02/22−rdf−syntax−ns#`>) which states a relation between a resource and its class.

RDFS also provides some primitives to describe relationships between classes or properties, and domain and range of a property. Some important resources in

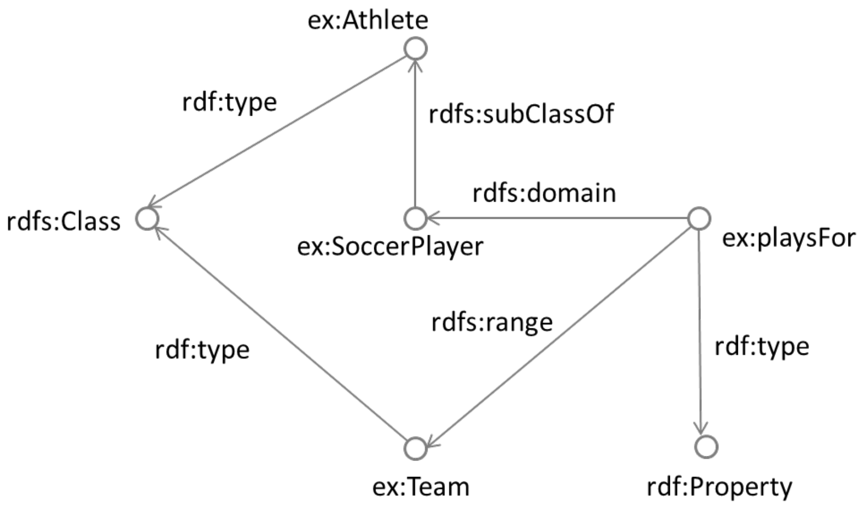


Figure 2.3: Example of RDF Schema.

RDFS are as follows (for brevity we use the "rdfs" prefix for
$< http://www.w3.org/2000/01/rdf\text{-}schema\# >$):

- *rdfs:Class.* Used to represent resources that are RDF classes.
- *rdf:Property.* Used to represent properties that are RDF properties.
- *rdfs:subClassOf.* Used as a predicate to mean that the subject is a subclass of the object.
- *rdfs:subPropertyOf.* Used as a predicate to mean that the subject is a subproperty of the object.
- *rdfs:domain.* Used as a predicate when the subject is a property and the object is the class that is domain of this property.
- *rdfs:range.* Used as a predicate when the subject is a property and the object is the class that is range of this property.

Example 2.2. *Figure 2.3 shows an RDF Schema describing a soccer player to be a subclass of the class athlete. It is possible to notice the property `ex:playsFor` (as instance of `rdf:Property`) with domain `ex:SoccerPlayer` and range `ex:Team`. The terms `ex:Athlete` and `ex:Team` are declared to be instances of the `rdfs:Class`.*

By using these relational primitives, the authors of an RDFS vocabulary implicitly define rules that allow additional information to be inferred from RDF graphs. For instance, consider the following triple `ex:SoccerPlayer`$_1$ `rdf:type ex:SoccerPlayer`. It is possible to infer `ex:SoccerPlayer`$_1$ `rdf:type ex:Athlete` because of the rdfs:subClassOf property. For more about the above features, we instead refer the interested reader to the RDF Schema [27].

Listing 2.2 illustrates the Turtle serialization of the Example 2.2.

Listing 2.2: Example of the property definition "ex:playsFor".

```
@prefix ex:   <http://example.org/ontology/> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema\#>

ex:playsFor rdf:type      rdf:Property .
ex:playsFor rdfs:domain   ex:SoccerPlayer .
ex:playsFor rdfs:range    ex:Team .
```

Web Ontology Language. Ontologies are the pillars of the Semantic Web as well as of Linked Data which specify the knowledge that is shared and exchanged between different systems. The knowledge specified is defined through the semantic of the terms used for describing data and the relations between these terms. Ontologies enable humans and machines to interpret the meaning of data that is being exchanged.

Web ontologies are ontologies that use one of the standard Web ontology languages and are published on the Web, like RDFS or OWL. OWL can be used to extend RDFS to formulate more expressive schema and subclass hierarchies, and additional logical constraints by enabling richer entailment[1] regimes. OWL is recognized by W3C Recommandation [13] and since 2008, OWL was extended to OWL2 which is also recognized by W3C [69].

In order to model knowledge about a domain of interest, OWL2 uses three ontology notions that are: axioms, entities and expressions[2]. An *axiom* is the elementary pieces that an OWL ontology can expresses and is often referred to as a statement. An example of statement is "Rome is the capital of Italy". In general, OWL statements can be either true or false given a certain state of affairs.

[1] Draw consequences from existing knowledge.
[2] Entities can be combined into expressions by creating complex descriptions.

In OWL terms: we say, a set of statements `A` *entails* a statement `a` if in any state of affairs wherein all statements in `A` are true, also `a` is true. There exist reasoners which can automatically compute entailments. In this way, axioms that are subtle and difficult for people to understand, are discovered by reasoners.

The constituent of statements in OWL2 are *individual names* (like Rome, Italy), categories denoted as *classes* (like city, state) or relations denoted as *properties* (like is capital of) [108]. In this thesis, we mainly work with individual names which comprises RDF terms that run in the subject or object position of a triple to denote objects of the real world. We use the term *entity* as a short form for an individual name.

A fact is either an instantiation (an axiom stating that an individual name has the type of a class expression), a relation, an attribute (uses almost the same form as relation) or an individual equality (an axiom stating that two names refer to the same individual). In particular, we consider facts as relations which have the form $\langle s, p, o \rangle$ with `s` and `o` being entities and `p` being a property. Intuitively, it means that the property `p` relating `s` to `o` *holds*, i.e. the fact is considered true under an interpretation.

When working with Linked Data, it is not necessary to faithfully replicate the complexity of the the OWL standard and its background. It is sufficient for the purposes of this thesis to understand that OWL goes beyond RDFS and bring a much richer semantics for use with RDF data. However, when reasoning over huge amounts of data, only the simplest reasoning processes are computationally efficient, and these can for the most part be implemented using only RDFS.

Query Language for RDF

SPARQL is the standard language for querying the RDF data model [120] developed by the W3C Data Access Working Group in 2008, as well as the SPARQL Protocol for formulating queries across diverse data sets through the Web. The results of SPARQL queries can be *results sets* or *RDF graphs.* We distinguish between graphs returned as answers to queries and graphs contained in documents by naming the former group of graphs as *answer graph.*

SPARQL queries contain one or more RDF graphs called *basic graph patterns* which in contrast to RDF triples of a data set, contains *triples patterns* where the subject, predicate or object position may be a variable. The triples in the basic graph pattern match against triples in the RDF data set thus producing a solution mapping, i.e. a result set. The result set is a set of bindings of variables to RDF terms. Each binding applied to the triple pattern returns a triple present in the RDF graph.

SPARQL depends on RDF and not on RDFS and OWL standards thus, it does not provide direct support for inferencing. But, in the next future, it is likely to be built an integration of RDFS and OWL entailment with SPARQL [52]. SPARQL language is similar to the database query language SQL with the difference that SPARQL is shaped by the fact that it operates over graph data represented as RDF triples, while SQL operates on tabular data organised in a relational database.

Example 2.3. *Consider we want to extract all the names and the appearances of the soccer players playing for the "AC Milan" team (see Listing 2.3).*

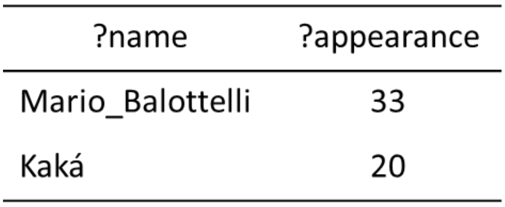
Listing 2.3: Example of a SPARQL query.

```
PREFIX dbo: <http://dbpedia.org/ontology/> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema\#>\\
PREFIX  :   <http://dbpedia.org/resource/> .

SELECT DISTINCT ?name ?appearance
FROM <http://dbpedia.org/data/AC_Milan.rdf>
WHERE {
  ?s rdf:type dbo:soccer_player;
     dbo:name  ?name;
     dbo:team  :AC_Milan
}
```

The query begins with `PREFIX` statements that define abbreviations for namespaces. The query begins in the line starting `SELECT`, which contains a variable starting with the question mark character '?'. This line specifies that we want to retrieve the name and the appearances of a soccer player. We can choose a name of the variable such that we use it consistently throughout the query. The remainder of the query, starting `WHERE`, contains a list of RDF triple patterns. The `WHERE` clause in the example has three RDF triple patterns, separated by a full stop. The first pattern matches resources of type soccer player; the second pattern states that the these resources has a name and the third pattern requires that these resources have object : AC_Milan.

The response to a query is computed by a process known as *graph matching*, shown diagrammatically in Figure 2.4, where both query and data set are shown as RDF graphs specified in Turtle (to simplify, only part of the above data set is included).



?name	?appearance
Mario_Balottelli	33
Kaká	20

FIGURE 2.4: Example of a result set of a SPARQL query.

A SPARQL query can be executed through a program or website that serves as a SPARQL endpoint. A SPARQL endpoint is an HTTP server (identified by a given URL) which receives requests from SPARQL clients.

2.1.3 Linked Data

Linked Data Principles

Linked Data (LD) can be considered as a specification of the Semantic Web which generates semantic connections among data sets. In 2006 Berners-Lee provided a set of principles for the interlinking and publication of structured data on the Web [22]. These principles are given as follows:

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

The first Linked Data principle stimulates the use of URIs to identify things. As in the Web of documents, in Linked Data a URI as a globally unique identification mechanism, is used to identify a document describing an entity. A document identified by a URI, which can return representations such as RDF descriptions is called an *information resource*[3] (IR). On the other hand, in Linked Data a URI identify not only documents but also real world objects or abstract concepts (places, people, images). The second principle advocates the use of the identification mechanism (URIs) through specific protocols such as the application level protocol, HTTP, to achieve interoperability between independent information systems. According to the third LD principle [22], we assume that each URI identifying an entity e is defererenceable. The dereferenceability of the URI entities rely on the HTTP mechanism, known as *content negotiation* [42]. This mechanism dereferences the URI that identifies the entity and returns the description of the entity in a specified data formats and language indicated by a user agent. For short we call a description of an entity, an entity document and we formally define it as follows:

Definition 2.4 (Entity document)**.** An *entity document* denoted as d^e with e being an entity, is the description of the entity which is returned after looking up the HTTP URI of the entity e.

Entity documents can be represented in the form of HTML pages when read by humans. Entity documents that are intended to be read by machines are represented as RDF documents in order to enable different applications to process the standardized content. Further, the applications should be aware of the difference between entities and their descriptions for further investigation (see Chapter 5. Some publishers (e.g., DBpedia) explicitly distinguish between a URI that denotes the entity and its description by assigning different URIs in order to be unambigous. For example, `http://www.example.org/resource/Milano` represents an entity and `http://www.example.org/data/Milano` represents the entity document of the resource Milano. Note that, the terms entity document and document will be used interchangeably in this work. Linked Data distributed across the Web applies a standard mechanism for specifying the connections between real world objects (fourth principles). The mechanism of interlinking is provided by RDF links that differently from the standard Web, RDF links connect entities and not just documents. The RDF links enable the process of discovering, accessing and integrating data in a straightforward way.

[3] All the essential characteristics can be conveyed in a message and be transported over protocols such as HTTP.

Linked Open Data

In 2007, the W3C Linking Open Data (LOD) project began publishing existing data sets under open licenses based on Linked Data principles. According to the open data definition[4], the data sets converted into RDF can be freely accessed, reused and redistributed.

LOD can be considered as a new application domain and because its importance has been growing over the last few years, it is important to study and analyse its quality. Along this way, to encourage people to publish good Linked Data, the initiator of the Linked Data paradigm, Tim Berners-Lee proposed a five-star rating system[5]. In this way data publishers can evaluate their data sets according to the following rating system:

- **One-Star (*)**: data is available on the web (whatever format), but with an open license.
- **Two-Star (**)**: data is available as machine-readable structured data (e.g., Microsoft Excel instead of a scanned image of a table).
- **Three-Star (***)**: data is available as (2) but in a non-proprietary format (e.g., CSV instead of Excel).
- **Four-Star (****)**: data is available according to all the above, plus the use of open standards from the W3C (RDF and SPARQL) to identify things, so that people can link to it.
- **Five-Star (*****)**: data is available according to all the above, plus outgoing links to other people's data to provide context.

Each additional star is promoted as increasing the potential reusability and interoperability of the publishers' data. The rating system measures the "quality" in terms of how much a data set conform to the Linked Data principles, while in general measuring the quality means evaluating a set of dimensions which capture specific aspects of data quality.

[4] http://opendefinition.org/

[5] http://opendefinition.org/http://www.w3.org/DesignIssues/LinkedData.html

2.2 Temporal RDF

In this section we define the core concepts of the temporal RDF data model, which extends triples and graphs with temporal information. In this thesis we regard time as a discrete, linearly ordered domain, as proposed in [58].

Temporal entities. We distinguish two types of temporal entities used for representing temporal information in RDF data: *time points*, represented by a single variable t_i which indicates a date, and *time intervals*, represented by the standard notation $[t_i{:}t_j]$ delimited by a starting time point t_i and an ending time point t_j where $t_i \leq t_j$.

Definition 2.5 (Disconnected time intervals)**.** Two time intervals $[t_i{:}t_j]$ and $[t_h{:}t_k]$ are *disconnected* iff $t_j < t_h$, or $t_k < t_i$, and *connected* otherwise.

In an RDF graph, a time interval associated with an RDF triple represents the time period when the triple was valid (i.e. the triple is considered true under an interpretation at a given time period). The validity of a triple can be bounded by a temporal context known as the valid time or the *temporal validity* of the triple [59]. The temporal context of a triple can be represented by temporal triples defined as follows:

Definition 2.6 (Temporal RDF triple)**.** Given a triple $\langle s, p, o\rangle$ and a temporal entity $[t_i{:}t_j]$ annotating the triple, a temporal triple is denoted as $(\langle s, p, o\rangle, [t_i{:}t_j])|t_i \leq t_j$.

We analogously define a temporal fact as a couple denoted as $(f, [t_i{:}t_j])$, where f is a fact and $[t_i{:}t_j]$ is a time interval that represents the temporal validity of the triple. We further refer to a temporal fact to as a *volatile fact* which denote facts that change over time.

In this work, we distinguish between a temporal RDF triple (for short temporal triple) and a sub-type of literal triples which we call *date literal triples*.

Definition 2.7 (Date literal triple)**.** A date literal triple, is a triple of the form $\langle s, p, t\rangle$ where the object t is a time point.

As an example, although DBpedia does not provide temporal triples, it provides date literal triples such as ⟨*ex:Kaka, ex:birthDate, "1985-02-05"^^xsd:date*⟩.

Definition 2.8 (Temporal RDF graph)**.** A *temporal RDF graph* is a graph G containing a set of temporal RDF triples.

Definition 2.9 (Temporal named graph)**.** A *temporal named graph* is a named graph u_G annotated with a temporal entity.

A *temporal annotation* is a temporal entity t annotating a triple or a graph. In the RDF data model a binary relation can be established between the triple or graph and the temporal entity through a property we call *temporal annotation property*[6]

Concrete representation of time points on the Web According to well-accepted best practices, time points are represented on the Web by means of *date formats*. RFC 2616 defines three different date formats that are used in the HTTP protocol[7]. The first *datetime* format, e.g., `Sun, 07 Sep 2007 08:49:37 GMT`, is defined by the standard RFC 822 [36] and is the most preferred. The second *datetime* format, e.g., `Sunday, 07-Sep-07 08:49:37 GMT`, is defined by the standard RFC 850 [76]. The third datetime format, e.g., `Sun Sep 7 08:49:37 2007`, is defined by ANSI C's *asctime* format. ISO 8601 defines a numerical date format [78]; an example of date according to this format is 2007-09-07T08:49:37.sZ. Based on this standard, dates can be also modelled as primitive datatypes in XML Schema [41]. The primitive types, `date`, `dateTime`, `gYearMonth`, `gYear`, `gMonthDay`, `gDay` and `gMonth` defined by these specifications are usually used in RDF data. An alternative representation of time for Linked Data, which denotes temporal entities with URIs and makes use of the OWL Time ontology [70] has also been proposed [35]. The OWL Time ontology represents through constructs the time, but there is still a need for mechanisms for the representation of the evolution of facts.

2.3 Information Quality

2.3.1 Defining Information Quality

In the last two decades, researchers and practitioners have shown an increasing interest on information quality issues. In particular, information quality has been addressed in different areas including management information systems, Web-based

[6] Adds temporal entities to data (i.e. triples or graphs).

[7] `http://www.ietf.org/rfc/rfc2616.txt`

information systems or data integration [49]. Note that, there is a distinction between data and information. Most definitions refers to *data* as to the "atomic" representation of real-world objects. While data typically lack content, meaning or intent, an *information* is a data with an interpretation. Although the distinction we made between "information quality" and "data quality" term, we will use them interchangeably in this work.

The concept of information quality, as all abstract concepts, has various definitions. A widely adopted definition given in the quality literature conceives information quality as "fitness for use" [82, 141, 87]. According to this definition, quality is considered a task dependent and as such a consumer will judge whether or not information fit to her task at hand. Further, the definition implies that quality is subjective since the quality of an information can be appropriate for one task but not appropriate for another task.

Information quality is a multifaceted construct that may depend on various factors such as accuracy, timeliness, completeness, relevancy, trustworthiness, understandability, consistency, conciseness, availability, or verifiability [141]. Each dimension captures a single aspect or a construct of information quality. In order to assess the quality of a data set used for a specific task, a consumer might include a set of quality dimensions. The quality dimensions are not independent of each other and together they can be combined to assess the quality in the context of its use.

2.3.2 Information Quality Assessment

Information quality assessment is defined as the process of evaluating if a piece of data meets the information consumers need in a specific use case [20].

Data quality assessment involves the measurement of *quality dimensions* that are relevant to the consumer. The dimensions can be considered as the characteristics of a data set. A data quality assessment *metric* or *measure* is a procedure for measuring a data quality dimension [20]. These metrics are heuristics that are designed to fit a specific assessment situation [96]. Since the dimensions are rather abstract concepts, the assessment metrics rely on quality *indicators* that allow for the assessment of the quality of a data source w.r.t the criteria [43].

Quality assessment can be both subjective and objective. Subjective measures involve measuring the quality dimensions that are relevant to the user and comparing the assessment results with the users quality requirements. One can use a questionnaire to measure consumer perceptions of data quality dimensions. Objective assessments can be task-independent or task-dependent. Task-independent metrics reflect states of the information without the contextual knowledge of the application, and can be applied to any data set, regardless of the tasks at hand. Task-dependent metrics are developed in specific application contexts.

2.3.3 Metadata for Supporting Quality Assessment

The process of measuring data quality is supported by quality related metadata as well as data itself. The metadata related to data quality plays an important role in supporting metrics evaluation since they store complementary information including data quality. Metadata often provide the information necessary to understand data and/or evaluate them [11].

Provenance as a particular case of metadata

With the growth of the Web and Web of Data consecutively [15, 20, 106], there has been an increasing need for metadata about data referred to as provenance. Provenance refers to the contextual metadata that provides details regarding the sources and their origins such as who created what, view of the full revision chain of the data, in case of data integration information about the original data sources, etc. According to a deep study provided from the W3C Provenance Incubator group[8] provenance importance is associated to three generic scenarios that encompass a high number of specific applications. Provenance gives consumers of the data clarity in a data integration process, trust and the terms under which it can be reused [56].

Provenance is not a new subject; there has been a lot of work about provenance in workflow systems, databases, knowledge representation and information retrieval. While the provenance approaches in the database context are considered managed by closed systems where there exist a full control of the data, in the open Web a

[8] http://www.w3.org/2011/prov/wiki/Main_Page

broader approach is required which consider the provenance of data sources coming from different systems [106].

Provenance representation should be independent of the technologies used. Thus, a conceptual data model that specifies the mechanisms that can be used to access provenance on the Web is proposed, referred to PROV[9]. PROV is a conceptual data model and as defined by W3C provenance group, it can be implemented in three serializations: RDF[10], PROV-XML[11] and PROV-N[12].

There exist other communities or vocabularies to represent provenance such as the Open Provenance ModelOpen Provenance Model (OPM), a community-driven provenance model that provides an alternative, more expressive vocabulary, that describes provenance in terms of Agents, Artifacts and Processes. Dublin Core[13] is another vocabulary for representing provenance, in particular, the most used properties are: `dc:creator`, `dc:publisher` and `dc:date`.

It is important to provide provenance in order to assess the quality of data such as trustworthiness. The assessment might be possible if trust is associated with the content or the provenance information of the data sources [79]. However, the concept of trust depends on how it is represented, calculated and used [6]. The author in [63] models trust based on provenance information such as meta-information about the publisher of the data set, creation method and creation time of the data set, and publisher and publication time of possible original sources (see Chapter 4).

Quality of Metadata

Metadata quality can be considered as a necessary condition for the quality of a data set. Metadata is considered to be an important element for the quality assessment of some dimensions, such as trustworthiness or Time-Related Quality Dimensions (Chapter 4). Some quality issues related to metadata regards mainly to completeness and accuracy [100]. Completeness of metadata refers to "the presence or absence of values in the metadata fields" and accuracy refers to "the

[9] http://www.w3.org/TR/prov-dm/

[10] http://www.w3.org/TR/prov-o/

[11] http://www.w3.org/TR/prov-xml/

[12] http://www.w3.org/TR/prov-n/

[13] http://dublincore.org/documents/dcmi-terms/

intellectual distance separating them from the true representation of the resource being described". For instance a study [128] showed that the records of a data set did not have all elements of Dublin Core vocabulary as was defined at the schema level or also incorrect use of the Dublin Core elements was found.

As in the case of data quality assessment there is also the need to asses metadata. A recent work presents an empirical study of the assessment of metadata quality applied in the context of education [113]. This empirical analysis proposes guidelines to support metadata annotators which can also be considered as a first step metadata improvement.

Note that, in this work we do not focus on assessing metadata quality but rather we focus on data quality assessment for which we propose techniques of data freshness assessment that deal with incomplete and inaccurate metadata (Chapter 6), and for the improvement phase we propose an enrichment of metadata based on an hybrid approach as explained in Chapter 7.

2.3.4 Organization of Quality Dimensions

In order to organise the information quality dimensions, the authors in [139] classify the data quality dimensions into four categories according to the type of information that is used as quality indicator:

- Intrinsic – are those dimensions that are independent of the user's context. They capture whether information correctly represents the real world and whether information is consistent in itself;
- Contextual – are those dimensions that highly depend on the context of the task at hand. In contrast to the intrinsic dimensions, contextual dimensions cannot be assessed in a general fashion but need to be assessed based on user's context and subjective preferences;
- Representational – those dimensions capture aspects related to the design of the data;
- Accessibility – the dimensions belonging to this category involve aspects related to the access and retrieval of data to obtain either the entire or some portion of the data for a particular use case.

Chapter 3

State of the Art

The work presented in this thesis focuses on the assessment of Time-Related Quality Dimensions in Linked Data. As mentioned also in the introduction section, Time-Related Quality Dimensions are the main quality dimensions in information systems. This section positions this thesis with respect to related works and it gives an overview of the state-of-the-art techniques of Time-Related Quality Dimensions assessment in information systems (Section 3.1), and in Linked Data (Section 3.2), at the intersection of which our work resides. Furthermore, as we already mentioned Time-Related Quality Dimensions are based on the availability and the quality of temporal meta-information. We propose related work that applies the extraction of temporal information which bound facts with temporal validity intervals (Section 3.3).

Time-Related Quality Dimensions also referred to as data freshness are usually captured by data currency and timeliness dimensions which are well known quality dimension in the literature of information quality [26, 141]. We limit the discussion of definitions about currency and timeliness since there will be a more detailed discussion for all LD quality dimensions in the next chapter (Chapter 4).

3.1 Assessment of Data Freshness in Information Systems

Assessing data freshness in traditional Database Management Systems (DBMSs) is a straightforward process due to the fact that DBMSs track data updates into log files. Data freshness can be therefore computed by retrieving the time of the last modification of the tuples. Since values in a tuple are not mutually independent, the freshness of a tuple can be considered equivalent to the freshness of the last edited tuple value. In contrast to DBMSs, the data model used in LD is different, the entity identifiers are independent from the triples where they occur, where the triples can be added or removed and where these actions are not represented in the data model. The above characteristics of the LD domain can be covered by a provenance model which is unknown in most cases.

Data freshness assessment plays an important role in an Data Integration System (DIS), which aims to consistently integrate data from different sources by solving conflicts between data. Because of data freshness assessment, it is possible to solve conflicts between the same data having different data freshness scores [2, 25, 116]. The authors in [25] present an analysis of data freshness definitions and measures according to the types of applications in DIS:

Data Warehousing Systems - estimate data freshness as the difference between delivery time and last update time (known as currency). Data freshness is provided in the design process based on the user expectations [133].

Caching Systems - estimate data freshness as the difference between the last modification time of an element of the database and the synchronization time (known as recency or age). Data freshness is established by the user preferences as a trade-off between freshness and latency according to their different applications [28].

Replication Systems - estimate data freshness as the time since the oldest data element has been waiting for the first refresh transaction (known as age).

Moreover, the method of assessing data freshness depends also on the frequency of change of data named as the “nature of data” and the synchronization policy adopted by the applications [25]. Several data freshness definitions have been proposed according to the variety of applications of DISs. However, the proposed definitions do not provide any formally noted measure to assess data freshness.

Henrich et al. [68] propose a new metric for quality assessment which is based on a probabilistic approach. Data freshness is given as the probability of a property value to be updated with respect to the real world property value at the time when data quality is assessed. More specifically, the probability is an exponential distribution of the age of a property value and the average decline rate for the given values of the property. There are two parameters in the probability approach that need to be calculated, that are, the age of the property value, which refers to the difference between the time when data quality is assessed and the time when data is acquired, and the decline rate, which refers to the average rate of the frequency of update of the property values of the property under consideration. In contrast, our techniques of data freshness assessment are domain independent, thus, do not require additional calculus, such as the case of decline rate. Furthermore, the work in [68] does not tackle the problem of the lack of availability of temporal annotations.

Other approaches tackle the problem of insufficient temporal annotations by estimating the age of a web resource [111, 123]. The approach in [111] estimates the last update of a Web document based on the last update of the neighbors Web documents. It considers three type of neighbors: incoming, outgoing and assets (e.g. images, objects, CSS files, etc) and for each retrieves an averaged last modification date. The neighbors taken under consideration are those that are close to the Web document where the closeness between the Web document and its neighbor is provided by a distance metric. Getting inspired by this approach, we propose a metric for data freshness of triples based on the data freshness of the entity documents which describe the entities occurring in the triple.

3.2 Assessment of Data Freshness in Linked Data

SIEVE is a framework for the evaluation of several quality dimensions, proposed in the context of LD, and applied to a data integration scenario [104]. The results show that data freshness is a crucial driver when data coming from heterogeneous sources have to be integrated and fused. SIEVE aims to be a general framework for representing several qualities; data freshness is measured in the experimental scenario, but the authors do not investigate the specific problem of measuring data freshness on arbitrary data sets. Our contribution can be seen as a point of support to the approaches such as SIEVE since the freshness evaluation needs temporal annotations of facts and documents, to be available.

The author in [65] measures timeliness according to [9] formula by using provenance information. The Provenance Vocabulary[1] is focused on representing metadata about the creation and access of data. Although this approach is a major step towards transparency of data quality in the Web of Data, we can currently not rely on the availability of previously created provenance information.

In this direction, [112] propose the modelling of provenance of DBpedia triples based on Wikipedia revision control mechanism. The provenance information, which indicates when and by whom a triple was created, is embedded in the fourth element of a quadruple represented by a URI. The URI contains information about the line, the section and the revision id of a Wikipedia article where the triple has been extracted from. In this way it is possible to retrieve the last modification date indicated in the revision of the Wikipedia page identified by the id. The approach retrieves provenance information and the last modification date of triples based on the revision control mechanism adopted by Wikipedia. This approach stores the last modification date in dumps and cannot be used on the fly by other applications.

3.3 Temporal Information Extraction

Several machine learning approaches have been developed to discover links between events and temporal information (e.g., dates) into one or more sentences of a document where the event is mentioned [136]. In alternative, the work in [77] presents a method to link events or facts with timestamps according to a classification approach. In contrast to these approaches, our approach is completely unsupervised and it does not need training data. Temporal Information Extraction (TIE) [98] is a more recent system that finds a maximal set of temporal annotations for events mentioned in a given sentence. Therewith, it can infer relations between these events using the temporal annotations. Instead of Allen-style intervals [3], TIE uses time points. However, this approach is not sufficient to extrapolate the temporal scope of facts because it focuses on the micro-reading of temporal annotations in single documents or sentences. Although the aim of temporal bounding [39] and our approach is the same since both retrieve temporal constraints given a fact, there are fundamental differences. NLP techniques employed in temporal bounding are more sophisticated but at the same time more

[1] http://trdf.sourceforge.net/provenance/ns.html

expensive and extract evidence from the text on a limited corpus. Our approach uses softer, but more efficient, NLP techniques to extract evidence from the whole web. Moreover, our approach investigates how to complement evidence retrieved from texts with evidence from the web of data.

Timely Yago2 [71] has the objective of enriching facts with temporal scopes. Instead of using the original data source (i.e, Wikipedia) where the link between facts and time intervals is explicitly made available, our approach exploit the evidence from the web of data where facts are not associated with time intervals and the web of documents, i.e., free text evidence. Yago2 identify the time of a fact if the time of the entities occurring in the fact is known and the property occurring in the fact belongs to a predefined category. PRAVDA [142] is a recently proposed method to harvest basic and temporal facts from free text. The approach is based on a semi-supervised label propagation algorithm that determines the similarity between structured facts and textual facts. Yet, it does not use the verbalization of RDF triples to check for RDF triples in text like DeFacto does. The system CoTS provided in [129] is similar to our system since it also detects temporal scopes for facts. In contrast to our approach, CoTS relies on document meta-data such as its creation data to assign temporal scopes to facts. To ensure that it gathers enough information, CoTS aggregates evidences from a large number of documents to temporally scope a set of facts. This approach is complementary to our current approach and can easily be combined with it.

Chapter 4

Linked Data Quality Dimensions

Assessment of quality is an important step for making consumers aware about the quality of the data sets that can be used in LD applications. The assessment process relies on the measurement of a set of data quality dimensions since data quality assessment is conceived as a multidimensional concept (see Chapter 2). Quality assessment is not a trivial task because data sets with quality problems might be useful for certain applications and not useful for other applications and thus may require users judgment or context involvement. Although there are a lot of methodologies, measures and tools proposed in other areas different from LD, quality assessment in LD is still in its infancy. Furthermore, LD presents new challenges that were not handled before in other research areas. Thus, adopting existing approaches for quality assessment in LD is not a straightforward problem.

We investigate data quality in LD and in particular, we present a set of dimensions that are a consequence of a long work of unification, formalisation and adoption of quality dimensions coming from three different areas: Web information systems, Semantic Web and relational databases.

The quality dimensions similarly to relational databases can be applied at data and schema level. The schema level refers to ontologies as the pillars of the Semantic Web and LD as well. Their assessment is also crucial for the re-usability of data. There have been efforts focused on evaluating the quality of ontologies [138, 47, 30]. Despite such recognized importance, the prevalent attention to the definitions of data quality dimensions goes to instances, which more extensively than schemas, are used in the Web of Data. As a consequence, in this chapter we deal especially with quality dimensions at data level rather than schema level.

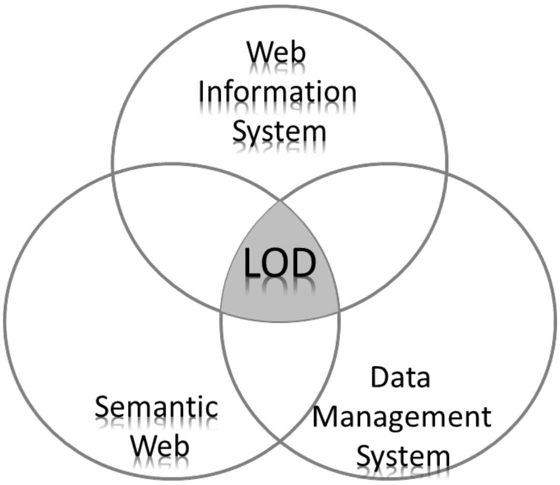


Figure 4.1: LD seen as a conglomeration of theories and technologies from three research areas.

We describe in detail data quality dimensions and their respective metrics, and for some dimensions we provide also an example.

The ideas presented in this chapter are the result of joint work with Universität Leipzig, Institut für Informatik, AKSW, Germany. The results of our joint work is under review as shown in Appendix A.

In Section 4.1 we provide a detailed description of quality dimensions applied in LD. Section 4.2 discusses inter relations between dimensions. To conclude we discuss some of the main open issues of data quality in LD (Section 4.3).

4.1 Quality Dimensions in LD

While there has been a lot of research on data quality in the past two decades, the topic has not yet received sufficient attention from the LD community.

LD can be consider as a conglomeration of theories and technologies from other areas such as the Semantic Web, the Web Information System and the relational database (see Figure 4.1). With the paradigm of LD, the Semantic Web standards can be effectively deployed on the Web in order to facilitate discovery and interoperability of structured data. The LD paradigm can be considered as a specification of the Semantic Web which generates semantic connections among data sets by providing links between entities. In this way, quality on the Web of Data includes a number of novel aspects, such as interlinking via links to external data sets, data representation quality or consistency with regard to implicit information. On the other hand due to its openness expressed by open access license, the LD paradigm share common characteristics with Web information systems [117] where data may often be used in ways not expected by the original creators of the LD data set (third-party application). Thus, some of the quality dimensions associated to the web pages, can be applied to the LD sources since the aim of both LD and Web information systems is to publish high quality information as soon as it becomes available. Finally, the structured nature of LD makes them similar to relational databases which handle large quantities of heterogeneous and distributed data. In contrast to relational databases that adopt the Closed World Assumption (CWA), LD follows the Open World Assumption (OWA) that has an influence on this discussion; OWA has an impact on the difficulty of defining and evaluating the compliance between data and schemas: a relation between two instances can hold even if the schema does not model such relation between the concepts the instances belong to; conversely, we cannot conclude that a relation between two concepts of different schemas does not hold because it is not represented in the data instances. The consideration about this assumption is out of the scope of this thesis and it is assumed that quality dimensions are treated as in a CWA.

Next section proposes a core set of different data quality dimensions, coming from the three different areas, that can be applied to assess the quality of LD. The identified dimensions are organized according to the classification introduced in Section 2.3.4 with the difference that Time-Related Quality dimensions are usually split between intrinsic (e.g., currency) and contextual (e.g., timeliness) group.

4.1.1 Intrinsic dimensions

Accuracy

Accuracy refers to the extent to which entities and facts are correct, that is, the degree to which they correctly represents the real-life phenomenon.

In LD accuracy similarly to relational databases can be classified into syntactic and semantic accuracy.

Syntactic Accuracy. *Syntactic accuracy* is defined as (a) the degree to which values correctly represent the domain values of the underlying vocabularies, and (b) the degree to which values conform to the syntax of its definition.

The validity of documents in [43] defined as "the valid usage of the underlying vocabularies and the valid syntax of the documents" is associated to syntactic accuracy. Fürber et al. [44] classified accuracy into syntactic and semantic accuracy. The authors explained that a "value is syntactically accurate, when it is part of a legal value set for the represented domain or it does not violate syntactical rules defined for the domain". Additionally, Hogan et al. [72] identify syntax errors such as RDF/XML syntax errors, malformed datatype literals and literals incompatible with datatype range, which we associate with syntactic accuracy.

Syntactic accuracy problems mostly refer to *literals incompatible with datatype range* or *malformed datatype literals*. For example, in the first case, a property `dbpedia:dateOfBirth` has the range `xsd:date` but we can find triples where `dbpedia:dateOfBirth` in the predicate position have `xsd:integer`. In the second case, consider that the dataype associated with a literal is `xsd:gYear` and we identify triples where `xsd:dateTime` literals are used instead. Other types of inaccuracies are related to properties or literals due to *misspelling errors* [72]. For instance, the object used in a triple is *Milano-Bicoca* instead of the correct one, *Milano-Bicocca*.

Syntactic incorrect values which do not comply to their definition in the underlying vocabulary can be assessed as the ratio between the number of syntactically incorrect values and the total number of values used in the data set, where syntactically incorrect values can be captured according to distance-based, deviation-based and distribution-based methods [20] or functional dependencies [44, 145]. As an example, consider John to be a correct property value of `foaf:name` and Jack an incorrect property value, then it is possible to evaluate the distance between the correct and the incorrect values by employing a *comparison function*. In alternative, some syntactic inaccuracies can be detected by enabling validators [43, 72] or by applying legal value rules (explicit definition of the allowed values for a certain property), legal value range rules (explicit definition of the allowed value range for properties holding numerical values) or syntactic rules (type of characters allowed and/or the pattern of literal values) [44]. To notice that syntactic accuracy could only be identified by comparing values in the data source with values in the underlying vocabularies.

Semantic Accuracy *Semantic accuracy* is defined as the degree to which data values correctly represent the real world facts.

Bizer [19] adopted the definition of accuracy from Wang et al. [139] as the "degree of correctness and precision with which information in an information system represents states of the real world". Furthermore, Furber et al. [44] classified accuracy into syntactic and semantic accuracy. He explained that values are semantically accurate when they represent the correct state of an object. Based on this definition, we also considered the problems of *spurious annotation* and *inaccurate annotation* (inaccurate labeling and inaccurate classification) identified in Lei et al. [95] related to the semantic accuracy dimension.

Semantic accuracy refers to accuracy of the meaning, i.e. facts can be either valid or non valid given a certain state of affairs (Section 2.1.2). The semantic accuracy problem in LD is determined between facts and their corresponding real world representations.

Let us consider for example an RDF document written in the Turtle syntax describing a set of triples from a data set as shown in the Listing 4.1.

Listing 4.1: Excerpt of an entity document for the resource `http://dbpedia.org/resource/Ronaldinho`

```
@prefix : <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix dbpprop: <http://dbpedia.org/property/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

:Ronaldinho dbo:deathDate ”2011-04-28”^^xsd:date .
:Ronaldinho dbpprop:currentclub   :Clube_Atl tico_Mineiro .
:Ronaldinho rdf:type   :Person .
```

In the first triple it is possible to observe a problem of semantic accuracy since the triple does not represent the status of the real world. This problem, known also as *spurious annotation*, shows how the triple cannot be mapped back to the real world object. In the third triple `:Ronaldinho` is classified as an instance of the class `:Person` rather than to a more precise class such as `:SoccerPlayer`. This problem represents the situation of an *inaccurate annotation* and in particular an *inaccurate classification* which means that the triple has been correctly represented but not accurately classified.

Semantic accuracy is more difficult to be assessed than syntactic accuracy because the vocabulary containing the definition of all terms in the syntactic accuracy is sufficient for the metric assessment. Here, there is the need of a real-world state representation which is usually given by a gold standard or a reference data set. However there are some metrics proposed in the literature such as: (a) validity of a fact that check the semantic accuracy of the fact against several sources or even several websites [93], (b) accuracy of the annotations, representation, labelling or classification is detected as a value between 0 and 1 [95]. Additionally, the correctness of the data set can be verified with the help of unbiased trusted third party (humans)[19].

Consistency

Consistency means that a knowledge base is free of (logical/formal) contradictions with respect to particular knowledge representation and inference mechanisms.

Bizer [19] adopted the definition of consistency from Mecella et al., [101] as when "two or more values do not conflict with each other". Similarly, Hogan et al. [72] defined consistency as "no contradictions in the data". Another definition was given by Mendes et al. [103] where "a dataset is consistent if it is free of conflicting information". Additionally, Böhm et al. [23] and Mostafavi et al. [107] present metrics to assess consistency. However, it should be noted that for some languages such as OWL DL, there are clearly defined semantics, including clear definitions what inconsistency means. In description logics, model based semantics are used: A knowledge base is a set of axioms. A model is an interpretation, which satisfies all axioms in the knowledge base. A knowledge base is consistent if and only if it has a model [8].

For example let consider the following facts in the Listing 4.1.

Listing 4.2: An example of axioms violating the dataset

```
:Boy owl:disjointWith :Girl .
:John a :Boy .
:John a :Girl .
```

The OWL property `owl:disjointWith` is used to state that the classes are disjoint and no instance can be at the same time an instance of both classes. However, after reasoning on the data set, it is possible to identify an inconsistency caused by the third triple since the instance John cannot be both a boy and a girl.

For assessing consistency, we can employ an inference engine or a reasoner, which supports the respective expressivity of the underlying knowledge representation formalism. In practice, Scalable Authoritative OWL Reasoner (SAOR) can be employed in order to shed light on the reasoning issues related to the interpretation of RDF data on the Web. Alternatively, RDF-Schema inference and reasoning with regard to the different OWL profiles can be used to measure consistency in a data set. Some inconsistencies issues can be given as follows:

- detection of use of entities as members of disjoint classes using the formula: $\frac{\text{no. of entities described as members of disjoint classes}}{\text{total no. of entities described in the data set}}$ [72];
- detection of misplaced classes or properties[1] using entailment rules that indicate the position of a term in a triple [72];
- detection of misuse of `owl:DatatypeProperty` or `owl:ObjectProperty` through the ontology maintainer[2] [72];
- detection of use of members of `owl:DeprecatedClass` or `owl:DeprecatedProperty` through the ontology maintainer or by specifying manual mappings from deprecated terms to compatible terms [72];
- detection of bogus `owl:InverseFunctionalProperty` values by checking the uniqueness and validity of the inverse-functional values [72];
- detection of the re-definition by third parties of external classes/ properties (ontology hijacking) such that reasoning over data using those external terms is affected [72];
- detection of negative dependencies/correlation among properties using association rules [23];
- detection of inconsistencies in spatial data through semantic and geometric constraints [107].

[1] For example, a URI defined as a class is used as a property or vice-a-versa.

[2] For example, attribute properties used between two resources and relation properties used with literal values.

Completeness

Completeness refers to the degree to which all required information is present in a particular data set. In terms of LD, completeness comprises of the following aspects: (a) Schema completeness, the degree to which the classes and properties of an ontology are represented, thus can be called “ontology completeness”, (b) Property completeness, metric of the missing values for a specific property, (c) Population completeness is the percentage of all real-world entities of a particular type that are represented in the data sets, and (d) Linkability completeness, which has to be considered especially in LD, refers to the degree to which instances in the data set are interlinked.

Bizer [19] adopted the definition of completeness from Pipino et al. [118] as “the degree to which information is not missing”. Fürber et al. [44] further classified completeness into: (a) Schema completeness, which is the degree to which classes and properties are not missing in a schema, (b) Column completeness, which is a function of the missing property values for a specific property/column, and (c) Population completeness, which refers to the ratio between classes represented in an information system and the complete population. Mendes et al. [103] distinguish completeness on the schema and the data level. On the schema level, a dataset is complete if it contains all of the attributes needed for a given task. On the data (i.e. instance) level, a dataset is complete if it contains all of the necessary objects for a given task. As can be observed, Pipino et al. provided a general definition whereas Fürber et al. provided a set of sub-categories for completeness. On the other hand, the two types of completeness defined in Mendes et al. can be mapped to the two categories (a) Schema completeness, and (c) Population completeness provided by Fürber et al.

As an example, let us consider in the Listing 4.3 a set of predicates relating two arguments of type *country* and *literal*:

Listing 4.3: The set of predicates relating two arguments of type *country* and *literal*.

```
:country  dbo:areaTotal  literal .
:country  dbo:populationTotal  literal .
```

The property completeness can be measured in terms of the proportion of the countries’ URIs (all European countries taken from DBpedia) having the above properties and the total number of countries found in the real world data set. Moreover, given the entity URI U, the reference entity URI $ref(U)$, represents the entities of the real world.

In general, completeness can be measured in terms of: (a) schema completeness - no. of classes and properties represented/total no. of classes and properties [19, 44, 103], (b) property completeness - no. of values represented for a specific property/total no. of values for a specific property [19, 44], (c) population completeness - no. of real-world objects are represented/total no. of real-world objects [19, 44, 103], (d) linkability completeness - no. of instances in the data set that are interlinked/total no. of instances in a data set [57]. It should be noted, that in this case, users should assume a closed-world-assumption where a gold standard data set is available and can be used to compare against the converted data set.

Linkability completeness is only considered in LD and not in relational data. Let us consider a set of resources and a network constructed for each of them. Optionally, a new set of edges can be added to each local network and a set of new local networks around the original set of resources is created. Once the original local network is created, an analysis of completeness of the link for each node is computed. In order to measure the completeness of links, we first need to assess the linkability dimension according to network measures such as linkability degree, cluster coefficient, owl:sameAs chains, centrality and description richness [57].

Conciseness

Conciseness refers to the minimization of redundancy of entities at the schema and the data level. Conciseness is classified into (a) intensional conciseness (schema level) which refers to the case when the data does not contain redundant schema elements (properties and classes), and (b) extensional conciseness (data level) which refers to the case when the data does not contain redundant objects (instances).

Mendes et al. [103] classified conciseness into schema and instance level conciseness. On the schema level (intensional), "a dataset is concise if it does not contain redundant attributes (two equivalent attributes with different names)". Thus, intensional conciseness measures the number of unique schema elements (i.e. properties and classes) of a dataset in relation to the overall number of schema elements in a schema. On the data (instance) level (extensional), "a dataset is concise if it does not contain redundant objects (two equivalent objects with different identifiers)". Thus, extensional conciseness measures the number of unique objects in relation to the overall number of objects in the dataset. This definition of conciseness is very similar to the definition of 'uniqueness' defined by Fürber et al. [44] as the "degree to which data is free of redundancies, in breadth, depth and scope". This comparison shows that uniqueness and conciseness point to the same dimension. Redundancy occurs when there are *equivalent* schema elements with different names/identifiers (in case of intensional conciseness) and when there are *equivalent* objects (instances) with different identifiers (in case of extensional conciseness) in a dataset.

Intensional conciseness measures the number of unique schema elements (i.e. properties and classes) of a data set in relation to the overall number of schema elements in a schema [103]. Extensional conciseness measures the number of unique entities in relation to the overall number of entities in the data set [103]. Further extensional conciseness can be measured as the total number of instances that violate the uniqueness rule in relation to the total number of relevant instances [44, 95]. In addition, representational conciseness can be measured as the detection of unambiguous annotations [95].

4.1.2 Contextual dimensions

Trustworthiness

Trustworthiness is defined as the degree to which the information is accepted to be correct, true, real and credible.

Trustworthiness is a crucial topic due to the availability and the high volume of data from varying sources on the Web of Data. Bizer [19] adopted the definition of trust from Pipino et al. [118] as "the extent to which information is regarded as true and credible". Jacobi et al. [79], similar to Pipino et al., referred to trustworthiness as a subjective measure of a user's belief that the data is "true". Gil et al. [50] used reputation of an entity or a dataset either as a result from direct experience or recommendations from others to establish trust. Additionally, Bizer [19] adopted the definition of objectivity from Pipino et al. [118] as "the extent to which information is unbiased, unprejudiced and impartial." Thus, reputation as well as objectivity are part of the trustworthiness dimension. Other articles [24, 45, 51, 53, 54, 62, 103, 126] provide metrics for assessing trustworthiness.

Trustworthiness can be measured by (a) computing triples trust values based on: provenance information which can be either unknown or a value in the interval [-1,1] where 1: absolute belief, -1: absolute disbelief and 0: lack of belief/disbelief [62], opinion-based method which uses trust annotations made by several individuals [51, 62] or provenance information and trust annotations in Semantic Web-based social-networks [53], (b) using annotations for data to encode two facets of information: blacklists (indicates that the referent data is known to be harmful) [24] or authority (a boolean value which uses the LD principles to conservatively determine whether or not information can be trusted) [24], c) using trust ontologies that assigns trust values that can be transferred from known to unknown data [79] using: content-based methods (from content or rules) or metadata-based methods (based on reputation assignments, user ratings, and provenance, rather than the content itself), d) computing trust values between two entities through a path by using: a propagation algorithm based on statistical techniques [126] or in case there are several paths, trust values from all paths are aggregated based on a weighting mechanism [126].

Relevancy

Relevancy refers to the provision of information which is in accordance with the task at hand and important to the users' query.

Relevancy is highly context dependent and is highly recommended in Web information systems since the process of retrieving the relevant information becomes complicated when dealing with a big flow of information.

Bizer [19] adopted the definition of relevancy from Pipino et al. [118] as "the extent to which information is applicable and helpful for the task at hand". Additionally, Bizer [19] adopted the definition for the amount-of-data dimension from Pipino et al. [118] as "the extent to which the volume of data is appropriate for the task at hand". Thus, since the amount-of-data dimension is similar to the relevancy dimension, we merge both dimensions. Flemming [43] defined amount-of-data as the “criterion influencing the usability of a data source”. While Pipino et al. provided a formal definition, Flemming and Chen et al. explained the dimension by mentioning its advantages.

The retrieval process of relevant data can be performed (a) using a combination of hyperlink analysis and information retrieval methods [19], (b) ranking (a numerical value similar to PageRank, which determines the centrality of RDF documents and facts [24]), (c) counting the occurrence of relevant data within meta-data attributes (e.g. title, description, subject) [19]. An alternative measure can be the coverage (i.e. number of entities described in a data set) and level of detail (i.e. number of properties) in a data set to ensure that there exists an appropriate volume of relevant data for a particular task [43].

Understandability

Understandability refers to the ease with which data can be comprehended without ambiguity and be used by a human information consumer.

This dimension can also be referred to as the comprehensibility of the information where the data should be of sufficient clarity in order to be used [19, 43]. Semantic Web was mainly planed to provide information in machine-readable format such that the data can be processed by machines. Understandability contributes towards the usability of machine-readable data from humans. In LD, data publisher are encouraged to provide human-readable labels and descriptions of entities.

Consider a data set about flight information given as follows:

Listing 4.4: Example of understandability.

```
 ex:m.049jnng    ex:departure    m.043j22x  .
 ex:m.049jnng    ex:arrival      m.045j23y  .
m.043j22x        ex:label      ``Boston Logan Airport''@en  .
```

The first two triples do not contain human-readable labels and thus the first two triples are not meaningful to the user besides the last triple that contains a human-readable label, saying that the entity `m.043j22x` refers to Boston Logan Airport.

Understandability can be measured as (a) the completeness of human-readable labelling of entities [40, 75], (b) the engagement of URIs[3] that follow a conventional

[3] http://www.w3.org/Provider/Style/URI

pattern [40], c) the availability of SPARQL queries examples [43], d) the ratio between all entities having one label and all entities with any label [40].

4.1.3 Accessibility dimensions

Licensing

Licensing is defined as the granting of permission for a consumer to re-use a data set under defined conditions.

Licensing is a new quality dimensions not considered for relational databases but mandatory in an open data world such as LD. Flemming [43] and Hogan et al. [75] both stated that in order to enable information consumers to use the data under clear legal terms, each RDF document should contain a license under which the content can be (re-)used. Additionally, the existence of a machine-readable indication (by including the specifications in a VoID[4] description) as well as a human-readable indication of a license are important not only for the permissions a license grants but as an indication of which requirements the consumer has to meet [43]. Although both these studies do not provide a formal definition, they agree on the use and importance of licensing in terms of data quality.

LD aims to provide users the capability of aggregating data from several sources, therefore the indication of an explicit license or waiver statement is necessary for each data source. Additional statements should be added to a data set clearly indicating the type of license or waiver or license details that should be mentioned in the VoID[5] file. [89] present a semantic framework for evaluating CC ShareAlike recursive statements. [137] present an approach, similar to what is proposed by [46], applied to the Web of Data scenario using Web languages only. They consider only CC licenses compatibility and composition.

A data set can choose a license depending on what permissions it wants to issue (e.g. restrictions, liability, responsibility). Possible permissions include reproduction, distribution or the modification and redistribution of data [105]. Providing licensing information increases the usability of the data set as the consumers or third parties are thus made aware of the legal rights and permissiveness under

[4] http://vocab.deri.ie/void

[5] http://vocab.deri.ie/void

which the pertinent data are made available. Licensing can be checked by the indication of machine and human readable information associated with the data set clearly indicating the permissions of data re-use.

Availability

Availability of a data set is the extent to which information (or some portion of it) is present, obtainable and ready for use.

Bizer [19] adopted the definition of availability from Pipino et al. [118] as "the extent to which information is available, or easily and quickly retrievable". Flemming [43] referred to availability as the proper functioning of all access methods. However, the definition by Pipino et al. is more related to the measurement of available information rather than to the method of accessing the information as implied in the latter explanation by Flemming.

Availability of a data set usually refers to retrieve the RDF description of an entity by dereferencing its HTTP URI. In addition it is possible to make data sets available through SPARQL endpoints or by downloading RDF dumps. LD search engines in alternative provide APIs for the crawled data.

Availability is measured by (a) checking whether the server responds to a SPARQL query [43], (b) checking whether an RDF dump is provided and can be downloaded [43], (c) detecting dereferencable URIs (by checking for dead or broken links [72], i.e. when an HTTP-GET request is sent, the status code `404 Not Found` is not returned [43], (d) useful data (particularly RDF) is returned upon lookup of a URI [72] and for changes in the URI, i.e. compliance with the recommended way of implementing redirections using the status code `303 See Other` [43]), and (e) detecting whether the HTTP response contains the header field stating the content type of the returned file such as `application/rdf+xml` [72].

Linkability

Linkability refers to the degree to which entities that represent the same concept are linked to each other, be it within or between two or more data sources.

Linkability is a new and relevant dimension in LD since it supports data integration and interoperability. Linkability is provided by RDF links that establish a relation between the entity identified by the subject and the entity identified by the object. Through the typed RDF links, data items are effectively interlinked. The importance of linkability, also know as "mapping coherence" can be classified in one of the four scenarios: (a) Frameworks, (b) Terminological Reasoning, (c) Data Transformation, (d) Query Processing, as identified in [102]. Moreover, it is important to ensure proper representation of the type of relationship between the entities, that is, the correct usage of the property (e.g. `owl:sameAs, skos:related skos:broader` etc.) [60].

In order to overcome these problems we need some metrics to assess the quality of links. The proposed metrics are based on network measures such as the linkability degree, cluster coefficient, owl:sameAs chains, centrality and description richness through owl:sameAs links [57]. In particular let consider the linkability degree which provides the number of incoming and outgoing edges (links) in a node (resource) [75]. It is not possible to answer if a node is good or not since we do not have a gold standard. Thus an ideal set of incoming and outgoing links is provided based on a power-law degree distributions. An effort towards assessing the quality of a mapping (i.e. incoherent mappings), even though no reference mapping is available, is provided in [102].

Performance

Performance refers to the efficiency of a system that binds to a large data set, that is, the more performant a data source is the more efficiently a system can process data.

Performance is a dimension that has an influence on the quality of the information system or search engine, however not on the data set itself. Flemming [43] states that "the performance criterion comprises aspects of enhancing the performance of a source as well as measuring of the actual values". Flemming [43] gave a general description of performance without explaining the meaning while Hogan et al. [75] described the issues related to performance. Moreover, Bizer [19], defined response-time as "the delay between submission of a request by the user and reception of the response from the system". Thus, response-time and performance point towards the same quality dimension.

Performance is measured based on high throughput: (maximum) number of answered HTTP-requests per second and scalability - detection of whether the time to answer an amount of ten requests divided by ten is not longer than the time it takes to answer one request [43]. Also, detection of usage of slash-URIs where large amounts of data is provided[6] [43]. Additional metrics are low latency[7]: (minimum) delay between submission of a request by the user and reception of the response from the system [19, 43].

Security

Security is the extent to which data is protected against alteration and misuse.

Flemming [43] referred to security as "the possibility to restrict access to the data and to guarantee the confidentiality of the communication between a source and its consumers". Additionally, Flemming referred to the verifiability dimension as the mean a consumer is provided with to examine the data for correctness. Bizer [19] adopted the definition of verifiability from Naumann et al. [110] as the "degree and ease with which the information can be checked for correctness". Without such means, the assurance of the correctness of the data would come from the consumer's trust in that source. It can be observed here that on the one hand Naumann et al. provided a formal definition whereas Flemming described the dimension by providing its advantages and metrics. Thus, security and verifiability point towards the same quality dimension i.e. to avoid alterations of the dataset and verify its correctness.

Security can be measured as the degree of using digital signatures to sign documents containing an RDF serialization, a SPARQL result set or signing an RDF graph [31, 43], or by verifying authenticity of the dataset based on provenance information such as the author and his contributors, the publisher of the data and its sources, if present in the dataset [43].

[6] http://www.w3.org/wiki/HashVsSlash

[7] Latency is the amount of time from issuing the query until the first information reaches the user [110].

4.1.4 Representational dimensions

Interoperability

Interoperability is the degree to which the format and structure of the information conforms to previously returned information as well as data from other sources.

Bizer [19] adopted the definition of representational-consistency from Pipino et al. [118] as "the extent to which information is represented in the same format". We use the term "interoperability" for this dimension. In addition, the definition of "uniformity", which refers to the re-use of established formats to represent data as described by Flemming [43], can be associated to the interoperability of the dataset. Additionally, as stated in Hogan et al. [75], the re-use of well-known terms to describe resources in a uniform manner increases the interoperability of data published in this manner and contributes towards the interoperability of the entire dataset.

In addition, different data sets of the same domain can represent concepts at different levels of the data structure. For instance, consider a data set `A` which represents currency values in `euro` and a data set `B` represents the same information in `dollars`, then the system has to deal with a representational issue.

Interoperability can be assessed by detecting whether the data set re-uses existing vocabularies or entities from existing established vocabularies. Re-use of well known vocabularies, rather than inventing new ones, not only ensures that the data is consistently represented in different data sets but also supports data integration and management tasks. In practice, for instance, when a data provider needs to describe information about people, FOAF[8] should be the vocabulary of choice. Moreover, re-using vocabularies maximises the probability that data can be consumed by applications that may be tuned to well-known vocabularies, without requiring further pre-processing of the data or modification of the application. Even though there is no central repository of existing vocabularies, suitable terms can be found in SchemaWeb[9], SchemaCache[10] and Swoogle[11]. Additionally, a comprehensive survey done in [125] lists a set of naming conventions that should

[8] `http://xmlns.com/foaf/spec/`
[9] `http://www.schemaweb.info/`
[10] `http://schemacache.com/`
[11] `http://swoogle.umbc.edu/`

be used to avoid inconsistencies[12]. Another possibility is to use LODStats [38], which allows to perform a search for frequently used properties and classes in the LOD cloud.

Representational-conciseness

Representational-conciseness refers to the representation of the data which is compact and well formatted on the one hand and clear and complete on the other hand.

Bizer [19], adopted the definition of representational-conciseness from Pipino et al. [118] as "the extent to which information is compactly represented".

Representational-conciseness can be measured as: (a) detection of long URIs or those that contain query parameters [75], or (b) detection of RDF primitives i.e. RDF reification, RDF containers and RDF collections [75].

The concise representation of data not only contributes to the human readability of that data, but also influences the performance of data when queried. Keeping URIs concise and human readable is highly recommended for large-scale and/or frequent processing of RDF data as well as for efficient indexing and serialization. Hogan et al. [75] associate the use of very long URIs (or those that contain query parameters) as an issue related to the representational conciseness of the data.

4.1.5 Time-Related Quality dimensions

Data sets in LD are considered to be dynamic[13]. Changes happen at both schema and instance level to reflect the real world state. Therefore, entity documents and links between entities can be added, removed or updated [134, 35]. If the change is not timely with respect to the real world state, the information consumed by users or applications become outdated. Out-of-date data may reflect invalid information (i.e. false information).

Time-Related Quality Dimensions (TRQD) capture important aspects of data regarding changes and updates in time (i.e. the dynamic nature of LD). Important aspects of time-related quality dimensions are data freshness over time (currency),

[12] However, they only restrict themselves to only considering the needs of the OBO foundry community but still can be applied to other domains

[13] http://www.w3.org/wiki/DatasetDynamics

the frequency of change over time (volatility) and data freshness over time for a specific task (timeliness). In the reminder of this thesis, we use the terms currency and timeliness, and data freshness interchangeably. While definitions of the three dimensions do not change with respect to the LD context, the metrics change according to the availability and diversity of temporal annotations represented in LD. In particular, temporal annotations are typically data- and application-specific.

Currency *Currency* concerns how promptly data are updated.

Consider an example where the triples are retrieved as of May 2013. The first and third triple in Listening 4.5 has high currency while the second triple does not represent a current result with respect to the real world at the time the data was retrieved. The object of the second triple is not updated to the current club which is `Parma`. In this case, currency is low.

Listing 4.5: The triples present some of the RDF data referring to the current clubs.

```
Ronaldinho         currentclub    Clube_Atl tico_Mineiro
Antonio_Cassano    currentclub    Internacional
Cristiano_Ronaldo  currentclub    Real_Madrid_C.F.
```

Currency of data elements is a surrogate measure used to indicate the whether data elements represents current values or not (i.e., valid values in the current state).

In practice, currency can be measured with respect to temporal annotations such as the *last update time*, which annotate documents or facts with time points or time intervals. Measuring currency of arbitrary documents or facts in LD presents several challenges as described initially in the introduction section and more details can be found in Chapter 6. As we already saw in Chapter 3, different methods are provided for measuring currency, which rely mainly on two components: (a) the time when the data was last modified, and (b) the observation time. According to these two components, currency of a data element is defined as time period between the two components. In Chapter 6, we adopt the same measure when temporal annotations are available and propose a propagation approach when temporal annotations are incomplete and inaccurate.

Volatility. *Volatility* characterise the frequency with which data is updated over time.

The authors in [25] presents three types of data elements, that is (a) *stable data elements* refers to data that does not change over time, (b) *long-term-changing data elements* refers, and (c) *frequently-changing data elements* refers to data that has intensive change. Since it is difficult to distinguish objectively between (b) and (c) we consider only two types of data elements: stable and volatile data elements over time. For example, the date of birth does not change and as such it can be classified as stable data. In Listing 4.5 the soccer players facts can be considered as frequently-changing facts, thus they become outdated and cease to be valid if the facts are not within the time period determined by volatility. Volatility of facts is a surrogate measure used to indicate the validity of facts. For example, if we notice that `Antonio Cassano` changes team on average every two years then we can deduce that after two years from the last change the fact is likely to be not valid.

Volatility is measured as the length of time during which data remains valid. This can be expressed by two components: (a) the expiry time (the time when the data becomes invalid), and (b) the input time (the time when the data was first published in LD). Due to the challenges we listed in currency, the expiry and input time are not always available. It is possible to adopt existing approaches that infer the change frequency of data elements at a given time [34] based on the change history of data elements. Notice that, the change history of facts in LD does not reflect the change of facts in the real world but the change of facts in the information system because LD sources are usually created offline or in a batch processing mode from Web sources or relational databases. The approach in [34] is based in the change history of data elements, which are not always available. In alternative, in Chapter 7 we propose an approach which does not only provide time intervals, but temporal validity of data elements, in particular, facts.

Timeliness *Timeliness* expresses how current data is for the task at hand.

The timeliness dimension is motivated by the fact that it is possible to have current data that is actually useless because it reflects a too old state of the real world for a specific usage. According to the timeliness dimension, data should ideally be recorded and reported as frequently as the source values change and thus never become outdated.

Timeliness is usually measured by combining currency and volatility. Although the combination of the two metrics is the approach in the right direction, this does not represent a necessary condition. Other factors that are not reflected in the system can influence the judgment about the timeliness of the information. For example, consider that we know the volatility of a soccer playing for a team and that the system records each change of the fact according to its volatility. In this way, the system assure the timeliness of the fact. Then, suppose that the player for some reasons cannot play and he leaves the team. In this case, the knowledge about the currency and volatility are not sufficient for assessing the timeliness in the system since this change is not reflected. On one side, the only to have knowledge about the real-world facts is the user who is able to assess the timeliness of the system.

However, differently from relation databases, LD can take advantage of the original data source and thus provide the assessment without the user's involvement (Section 6.5.2). As we observed in volatility, data sets in LD usually do not directly represent facts from the real world but from an existing information system which in turn is considered as the real-world representation. For example, consider we have the same information represented in two different data sets `A` and `B`. We want to link data from our data set to one of these data sets. In order to do that we first need to check which of the two sources is timely with respect to the real-world data. A first metric measures the delay between a change in the relational database and the respective change in the data set `A` (similar with the data set `B`). In this way, we can decide which data set is more timely and then connect to our data.

4.2 Inter-relationships between dimensions

The data quality dimensions explained in the previous section are not independent from each other but correlations exist among them. If one dimension is considered more important than the others for a specific application (or use case), then the choice of favoring it may imply negative consequences on the others. Investigating the relationships among dimensions is an interesting problem, as shown by the following examples of the possible interrelations between them. In this section, we describe the intra-relations between some of the dimensions.

First, consider the relationship between trustworthiness, semantic accuracy and timeliness. When assessing the trust of a LD data set, the semantic accuracy and the timeliness of the data set should be assessed. Frequently the assumption is made that a publisher with a high reputation will produce data that is also semantically accurate and current, when in reality this may not be so.

Second, relationships occur between timeliness and the semantic accuracy, completeness and consistency dimensions. Indeed, having semantically accurate, complete or consistent data may require time and thus timeliness can be negatively affected. Conversely, having timely data may cause low accuracy, incompleteness and/or inconsistency. Based on quality preferences given by web application, a possible order of quality can be as follows: timely, consistent, accurate, and then complete data. For instance, a list of courses published on a university website might be first of all timely, secondly consistent and accurate, and finally complete. Conversely, when considering an e-banking application, first of all it might be accurate, consistent and complete as stringent requirements and only afterwards timely since delays are allowed in favour of correctness of data provided.

The representational-conciseness dimension (belonging to the representational group) and the conciseness dimension (belonging to the intrinsic group) are also closely related with each other. On the one hand, representational-conciseness refers to the conciseness of *representing* the data (e.g. short URIs) while conciseness refers to the compactness of the *data itself* (redundant attributes and objects). Both dimensions thus point towards the compactness of the data. Moreover, representational-conciseness not only allows users to understand the data better but also provides efficient process of frequently used RDF data (thus affecting performance). On the other hand, Hogan et al. [75] associated performance to the issue of "using prolix RDF features" such as (a) reification, (b) containers and (c) collections. These features should be avoided as they are cumbersome to be represented in triples and to be expensive to support in data intensive environments.

Additionally, the interoperability dimension (belonging to the representational group) is interrelated with the consistency dimension (belonging to the intrinsic group), because the invalid usage of vocabularies may lead to inconsistency in the data.

There exists an inter-relation between the conciseness and the relevancy dimensions. Conciseness frequently positively affects relevancy since removing redundancies increases the proportion of relevant data that can be retrieved.

The linkability dimension is associated with the syntactic accuracy dimension. It is important to choose the correct similarity relationship such as *same, matches, similar* or *related* between two entities to capture the most appropriate relationship [60] thus contributing towards the syntactic accuracy of the data. Additionally, linkability is directly related to the linkability completeness dimension. However, the interlinking dimension focuses on the quality of the interlinks whereas the interlinking completeness focus on the presence of *all* relevant interlinks in a data set.

These examples of inter-relations between the dimensions, belonging to different groups, indicate the interplay between them and show that these dimensions are to be considered differently in different data quality assessment scenarios.

4.3 Summary

In this chapter, we have presented a list of quality dimensions applied to LD. The goal of this work is to obtain a clear understanding of the quality dimensions and metrics for LD quality assessment. We analyzed the dimensions and metrics with respect to previous works, in particular to relational databases.

As this study reveals, most of the dimensions preserve the same definition but not the method of measuring their respective metrics. The metrics depends a lot on the data input which provides new challenges.

The number of publications published regarding quality in LD in the span of 10 years is rather low and the reason may regard to the infancy of the research area or the possible re-use of research from mature, related domains. Additionally, in most of the existing works, the metrics were often not explicitly defined or did not consist of precise statistical measures.

Meanwhile, there is much research on data quality being done and guidelines as well as recommendations on how to publish "good" data. However, there is less focus on how to use this "good" data. The quality of data sets should be assessed and an effort to increase the quality aspects that are improper should be performed thereafter. We deem our data quality dimensions to be very useful for data consumers in order to assess the quality of data sets. As a consequence, query answering can be increased in effectiveness and efficiency using data quality criteria [110].

Chapter 5

Temporal Information in Linked Data

As we saw in Section 4.1.5, a prerequisite for being able to assess TRQD is to acquire temporal annotations. The acquisition of temporal annotations is a difficult task because time introduces a further dimension to data which cannot be easily represented in RDF, a language based on binary relations; as a result, several approaches for representing temporal annotations in Linked Data have been proposed.

Temporal annotations can be considered as a specialization of temporal information that we name temporal meta-information. Temporal meta-information is particularly relevant to several application domains because it annotates RDF statements and graphs with information about their creation, modification and validity. This distinction is relevant for further exploration and understanding of the approaches adopted for representing temporal meta-information.

In general, we investigate temporal information and in particular temporal meta-information published in Linked Data. We analyse the availability and characterisation of temporal meat-information both from a qualitative and quantitative perspective. We discuss the adoption of the approaches for representing temporal meta-information in terms of advantages and shortcomings based on the literature review and the experimentations in a large variety of data sets in the LOD cloud. Since the analysis of the whole LOD cloud is unfeasible, we use the large Billion Triple Challenge[1] (BTC) data set for our investigation.

[1] http://km.aifb.kit.edu/projects/btc-2011/

This chapter is organized as follows: Section 5.1 describes the relevance of the problem; in Section 5.2 we introduce the notion of temporal information and temporal meta-information. In Section 5.3, we review the approaches proposed in the literature for the representation of temporal meta-information and discuss their adoption in well-known data sets. In Section 5.4 we conduct experiments to quantitatively investigate the availability of temporal information in general, and in particular the adoption of the representation approaches of temporal information in the LOD cloud using the BTC data set and we discuss our findings. In Section we provide recommendations for data publishers and consumers 5.5. In Section 5.6, we draw the conclusions.

5.1 Overview

As the information on the Web can change rapidly [33], also Linked Data on the Web cannot be assumed to be static, with RDF triples frequently added to and removed from published data sets [83]. As a consequence, change management and temporal information are receiving an increasing attention in the LD domain. In particular, a number of significant issues have been investigated: a resource versioning mechanism for Linked Data, which allows for publishing time-series of descriptions changing over time [37]; a method to monitor the published data sets, successfully applied to several sources [84]; the maintenance of links over evolving data sets [119].

In *Semantic Data Integration*, time information can be used to assess the currency of information in order to favor the most up-to-date information when fusing data [104, 114]. The capability of managing time information plays a crucial role not only in the assessment of TRQD for the reasons explained in Section 4.1.5, but also in other applications and research areas. The analysis of time information can also support entity resolution in some complex scenarios where the values of the attributes considered in the matching process change over time [97]. In *Temporal Query Answering and Search*, temporal information can be used to filter out the data of interest given some temporal constraint, or to rank the results of a search engine on a temporal basis. Timelines associated with data can improve the *User Experience* by presenting information in a time-dependent order [143, 4].

The capability of designing effective solutions for the aforementioned applications depends on the availability of temporal information and the possibility to collect and process this information across heterogeneous data sets. For example, the modification date associated with RDF documents and extracted via HTTP protocol analysis has been used to fuse data coming from different DBpedia data sets [104]; however, this information is not available in many data sets. Understanding the current status of temporal information published as Linked Data is fundamental for the development of techniques used for the assessment of TRQD. However, this analysis has a broader scope, given the importance that temporal information play in other research areas.

Several approaches are provided to represent and query temporal information in RDF [58, 35, 143, 88], support versioning for Linked Data [119], and monitor changes [134, 84]. Although there have been several approaches for representing temporal information, there is still little understanding about the actual practice of using these representations. Recently, a study in [124] showed an empirical evaluation of the modelling patterns of temporal information based on a user-centric perspective, where users with different level of expertise in the area have to select the modelling pattern for a given modelling problem. However, our study provides a large-scale analysis to investigate the availability of the approaches adopted in Linked Data. To the best of our knowledge a systematic and large-scale analysis in this field was missing. The ideas presented in this chapter are the result of a joint work with Karlsruhe Institute of Technology, AIFB (Germany), that have been published as shown in Appendix A.

5.2 Temporal Information and Temporal Meta-Information

In this section, we first introduce our terminology and present a classification of the approaches used for representing temporal information. Given that the terms *entity*, *fact*, *entity document* and *temporal entity* are already defined in Chapter 2: we define a temporal information and a temporal meta-information as follows.

Definition 5.1 (Temporal information)**.** A *temporal information* refer to a ternary relation $T(x, a, t)$, where x is an entity, a fact, or an entity document, a is a property symbol, and t is a temporal entity.

We call *temporal property* any property used in a temporal information.

Definition 5.2 (Temporal meta-information)**.** A temporal information $T(x, a, t)$ is a *temporal meta-information* if and only if the temporal property a is a temporal annotation property that links a fact or an RDF document x with a temporal entity t.

As we already explained in Section 2.2, a temporal annotation property is a temporal property that is distinguishable in that it links temporal entities with data where by data we mean facts and RDF documents.

Temporal meta-information is further distinguished in: *versioning metadata*, which refers to temporal annotations that annotate facts or documents (usually documents) with time points through temporal property annotations such as the *last modification time*, *update time*, *extraction time* or *creation time*; and *temporal validity metadata*, which refers to temporal annotations that annotate facts or documents (usually facts) with time intervals. Therefore, the concept of temporal meta-information makes a clear distinction between generic temporal information that annotates entities with temporal information such as the birth date of a person or also the creation date of a PDF document, and the specific case of temporal information where facts and entity documents are annotated, e.g, the temporal validity of a fact, or the last update of an RDF document. A possible way to make it distinguishable and interpretable from the machines, is to make URIs for entities distinguishable from URIs for entity documents. For example the creation date of a book may be rather different to the creation date of a document that describes this book. In the following section we provide a qualitative analysis of the approaches adopted to represent temporal-meta information, keeping the distinction between versioning metadata and temporal validity.

5.3 Qualitative Analysis of Temporal Meta-information

Because of the tight constraints given by the triple-based structure of RDF descriptions, the concrete RDF-based representation of an even simple temporal meta-information requires some sophisticated mechanisms. Several approaches for providing a concrete representation of the temporal meta-information have been proposed. In Figure 5.1 we propose a classification of the different approaches of temporal meta-information representation in RDF where some of them adopts modelling patterns such as N-ary relationship. In Figure 5.1 we may identify

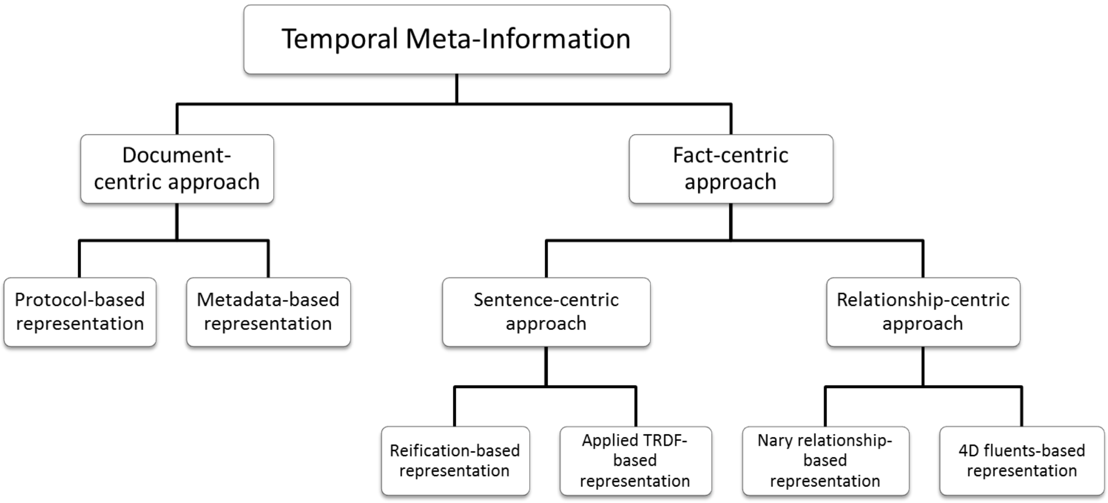

Temporal Meta-Information
Document-centric approach
Fact-centric approach
Protocol-based representation
Metadata-based representation
Sentence-centric approach
Relationship-centric approach
Reification-based representation
Applied TRDF-based representation
Nary relationship-based representation
4D fluents-based representation

FIGURE 5.1: Classification of approaches representing temporal information.

two core approaches that have been adopted for the concrete representation of temporal meta-information:

- Document-centric approach, where time points are associated with RDF documents, aims to provide versioning metadata.
- Fact-centric approach, where time points or intervals (usually intervals) are associated with facts, aims to prove temporal validity of facts; since facts can be represented by one or more triples - we further separate the Fact-centric approach into:
  - Sentence-centric approach, which explicitly define the temporal validity of one or more facts annotating them with time points or intervals.
  - Relationship-centric approach, which encapsulates time points or intervals into objects representing n-ary relations.

In the following we explain in detail the aforementioned approaches.

5.3.1 Document-centric approach

RDF documents, can be associated with temporal entities following two approaches. The first approach implicitly expresses temporal meta-information through the uses of HTTP-metadata, and in particular the Last-modified field of the HTTP response header. The second approach expresses temporal meta-information explicitly by using RDF triples with temporal annotation properties taken from available vocabularies such as Dublin Core[2]. Note that the *Last-modified* and *ETage* temporal annotation properties of HTTP headers used in the temporal meta-information of the Document-centric approach have been used also for the detection of changes in Web documents publishing RDF data [134].

Protocol-based representation. A Protocol-based representation adopts point-based time modelling; the temporal entity is not persistently associated with a Web document, but can be extracted from the HTTP header returned in response to an HTTP GET request for the document. The Protocol-based representation associates a time point, represented by a date, with a Web document using a predicate a (e.g., `Last-Modified`) defined in the HTTP protocol according to the schema as shown in Listing 5.1:

Listing 5.1: Protocol-based representation.

```
HTTP Response Header
 Status: HTTP/1.1 200 OK
 $a$ : $t_i$
```

Metadata-based representation. Let $\langle s, p, o \rangle$ be a fact, u_G a named graph (introduced in Section 5.2), a_G a temporal annotation property, t_i a time point; the Metadata-based representation associates a temporal entity with an RDF document as shown in Listing 5.2:

Listing 5.2: Metadata-based representation.

```
$\langle s, p, o, u_G \rangle$
$\langle u_G, a_G, t_i, u_G \rangle$
```

Figure 5.2 shows that the entity document describing the resource `Antonio Cassano` is modified in date `2011-08-30`.

[2] http://dublincore.org/documents/dcmi-terms/

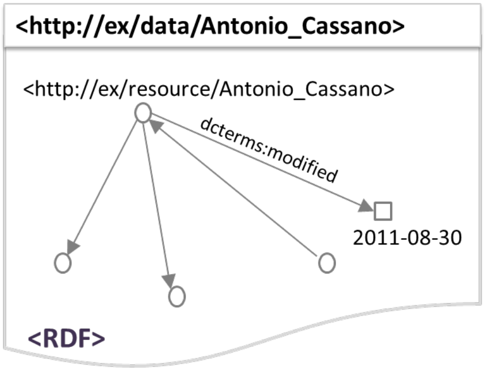

<http://ex/data/Antonio_Cassano>
<http://ex/resource/Antonio_Cassano>
dcterms:modified
2011-08-30
<RDF>

FIGURE 5.2: Metadata-based representation.

Application examples of data sets providing temporal meta-information having documents at subject position are: Protein knowledge base (UNIPROT[3]) and legislation.gov.uk.

5.3.2 Fact-centric approach

In the Fact-centric approach facts are associated with temporal entities that constrain their valid time. The first RDF model proposed to formally capture this idea is Temporal RDF [58]. In this model, RDF triples are annotated with temporal validity intervals.

In Section 2.2, we introduced temporal triples which cannot be encoded into the triple-based RDF data model because RDF can "natively" represents only binary relations. In order to solve this problem, several approaches that annotate facts with temporal entities in the standard RDF syntax have been proposed. These approaches follow two approaches that present significant differences: the Sentence-centric approach and the Relationship-centric approach.

Sentence-centric approach

Two strategies are adopted to represent the valid time interval of facts adopting the Sentence-centric approach.

[3] http://www.uniprot.org/

Reification-based representation. Let $\langle s, p, o\rangle$ be a triple, s^{st} an identifier of a triple, a_i^S and a_j^S two temporal properties, and $[t_i{:}t_j]$ a time interval; a Reification-based representation is defined as shown in Listing 5.3:

Listing 5.3: Reification-based representation.

$\langle s^{st}$, rdf:type, rdf:Statement$\rangle$
$\langle s^{st}$, rdf:subject, $s\rangle$
$\langle s^{st}$, rdf:predicate, $p\rangle$
$\langle s^{st}$, rdf:object, $o\rangle$
$\langle s^{st}, a_i^S, t_i\rangle$
$\langle s^{st}, a_j^S, t_j\rangle$

The first four sentences encode the reification of the triple representing the fact using the RDF vocabulary. The temporal annotation properties a_i^S and a_j^S link the triples respectively to the starting and the ending point of the time interval $[t_i{:}t_j]$ associated with the fact. Notice that a temporal annotation property a^S can have a time point or a time interval as property value.

Figure 5.3 shows on the left hand side a temporal triple (⟨`ex:Antonio_Cassano, ex:palysFor,ex:Milan` ⟩,[2011:2012]) which is reified as shown on the right hand side of the figure. In this case, the triple identified by a blank node is associated with a temporal validity represented by a time interval [2011:2012] through the property `validTimeInterval`. Instead of using a time interval, it is also possible to use two date literal triples indicating respectively the starting and the ending time point of the time interval. The constituent of the triple are represented as three other statements through the use of properties of the reification vocabulary which link the blank node with the subject `ex:Antonio_Cassano`, the property `ex:playsFor` and the object `ex:Milan` of the reified triple.

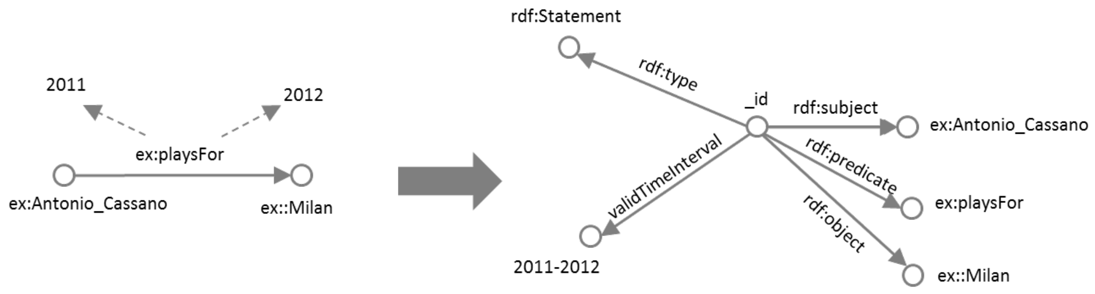


FIGURE 5.3: Reification-based representation.

Application example of a data set providing temporal meta-information according to Reification-based representation is Timely Yago [143].

In the above approach, every sentence associated with a temporal entity has to be reified. An alternative modelling pattern allows grouping together facts that have the same temporal validity interval by introducing the concept of *temporal named graph* [131]. As was introduced in Section 2.2, temporal RDF graphs are named graphs annotated with time intervals; each time interval is represented by exactly one temporal graph, where all triples belonging to this graph share the same valid time period. Temporal meta-information are collected in a *default graph* which occur as context in the quads as explained in Section 5.2.

Applied Temporal RDF-based representation. Let u_{TG} and u_G be the names respectively of a temporal named graph and of the default graph, a_i^S and a_j^S two temporal annotation properties, $[t_i{:}t_j]$ a time interval and $\langle s, p, o\rangle$ a triple; the Applied temporal RDF-based representation is defined as shown in Listing 5.4:

Listing 5.4: Applied Temporal RDF-based representation.

$\langle u_{TG}, a_i^S, t_i, u_G\rangle$
$\langle u_{TG}, a_j^S, t_j, u_G\rangle$
$\langle s, p, o, u_{TG}\rangle$

The temporal annotation properties a_i^S and a_j^S link the temporal graph respectively to the starting and the ending point of the time interval $[t_i{:}t_j]$. More triples can be associated with the same temporal graph.

Figure 5.4 shows on the left hand side a temporal triple (⟨`ex:Antonio_Cassano,` `ex:palysFor,ex:Inter` ⟩,[2012:2013]) which is represented in the temporal RDF data model as shown on the right hand side of the figure. In this case, the approach enables the use of temporal named graphs such as U$_{TG1}$ and U$_{TG2}$ each annotated with a temporal validity interval [2011:2012] and [2012:2013] respectively. The statement having the time interval [2012:2013] will belong to the temporal named graph U$_{TG2}$ with the same temporal validity.

Application example of a data set providing temporal meta-information according to Applied Temporal RDF-based representation is EvOnt [131].

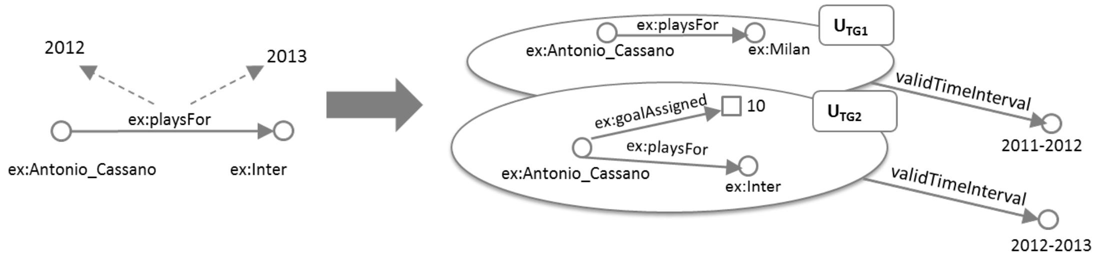


Figure 5.4: Applied Temporal RDF-based representation.

Relationship-centric approach

N-ary Relationship design patterns[4] are introduced to represent RDF relations with arity greater than two. These patterns model an n-ary relation with a set of RDF triples by (i) introducing a specific resource to identify the relation, and (ii) creating links between this resource and the constituents of the relation (resources and literals). These patterns can be used to associate temporal entities with facts represented by RDF triples to constrain their valid time. For example, the fact "Alessandro Del Piero (ADP) plays for Juventus", which is valid within the time interval [1993:2012], can be modelled as a quintuple ⟨ADP, playsFor,Juventus,1993,2012⟩ and represented following the N-ary Relationship pattern. A resource r is introduced to identify the relation and the temporally annotated fact can be represented by the set of RDF triples ⟨ADP,playsFor,r ⟩, ⟨ r,team,Juventus⟩, ⟨ r,from,1993⟩, ⟨ r,to,2012⟩. The direction of the links and the strategies adopted for naming the properties can change according to different variants of the pattern [88, 122]. However, the temporal entities are linked to the entities that identify a relation in all the proposed variants. In this thesis we define the N-ary Relationship-based representation adopting the variant described in the second use case of the W3C document, the one that occurs more frequently in the BTC corpus.

N-ary-relationship-based representation. Let $\langle s, p, o\rangle$ be a fact representing a relation p between s and o, r a new resource, p_1 and p_2 two properties, a_i^R and a_j^R two temporal annotation properties, and $[t_i{:}t_j]$ a time interval; the N-ary-relationship-based representation is defined as shown in Listing 5.5:

Listing 5.5: N-ary-relationship-based representation.

$\langle s, p_1, r\rangle$

[4] http://www.w3.org/TR/swbp-n-aryRelations/

$\langle r, p_2, o\rangle$
$\langle r, a_i^R, t_i\rangle$
$\langle r, a_j^R, t_j\rangle$

Although p_1 and p_2 can be two new properties, one of the two is usually equal to p as in the example discussed above.

Figure 5.5 shows on the left hand side a temporal triple (⟨`ex:Antonio_Cassano, ex:palysFor,ex:Milan` ⟩,[2011:2012]) which is represented in the temporal RDF data model as shown on the right hand side of the figure. In this case, the approach introduces a new class for the property `playsFor` and create instances of that resource, that is, `_:teamRelation` to relate to other resources, that are, the subject ex:Antonio_Cassano and the object `ex:Milan` of the original triple on the left hand side, and then to the begging time point 2011 and to the ending time point 2012.

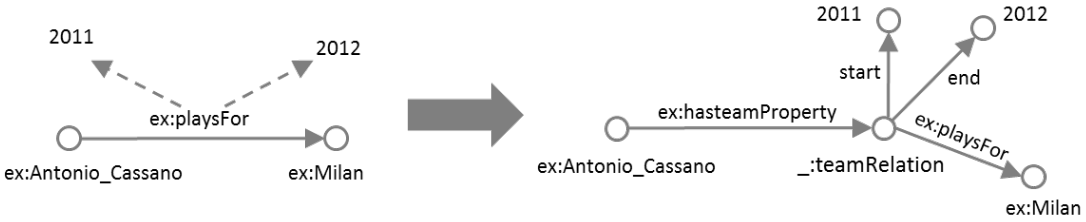


Figure 5.5: N-ary-relationship-based representation.

Application example of a data set providing temporal meta-information according to N-ary-relationship-based representation is Freebase[5].

A second representation approach of temporal meta-information according to the Fact-centric approach is based on the concepts of *fluents* and *timeslices* applied to RDF [144]. Fluents are properties that hold at a specific moment in time, i.e., object properties that change over time. The properties representing fluents link two timeslices, i.e., entities that are extended through temporal dimensions.

4D-fluents-based representation. Let $\langle s, p, o\rangle$ be an RDF triple, a_i^R and a_j^R two temporal annotation properties, $[t_i{:}t_j]$ a time interval, and s^t and o^t two timeslices associated respectively with s and o; the 4D-fluents-based representation is defined as shown in Listing 5.6:

[5] http://www.freebase.com/

Listing 5.6: 4D-fluents-based representation.

$\langle s^t, \mathrm{rdf{:}type}, \mathrm{:TimeSlice} \rangle$
$\langle s, \mathrm{:hasTimeslice}, s^t \rangle$
$\langle s^t, a_i^R, t_i \rangle$
$\langle s^t, a_j^R, t_j \rangle$
$\langle o^t, \mathrm{rdf{:}type}, \mathrm{:TimeSlice} \rangle$
$\langle o, \mathrm{:hasTimeslice}, o^t \rangle$
$\langle o^t, a_i^R, t_i \rangle$
$\langle o^t, a_j^R, t_j \rangle$
$\langle s^t, p, o^t \rangle$

Figure 5.6 shows the temporal triple (⟨`ex:Antonio_Cassano, ex:palysFor,ex:Milan` ⟩,[2011:2012]) which is represented according to the 4D-fluents-based representation. Two entities `ex:Antonio_Cassano` and `ex:Milan` are considered static (their value do not change in time), while the property `playsFor` is dynamic whose value may change in time. Because it is a *fluent* property, its domain and range is of class `Timelice`. `ex:Antonio_Cassno`^{T} and `ex:Milan`^{T} are instances of class `TimeSlice`.

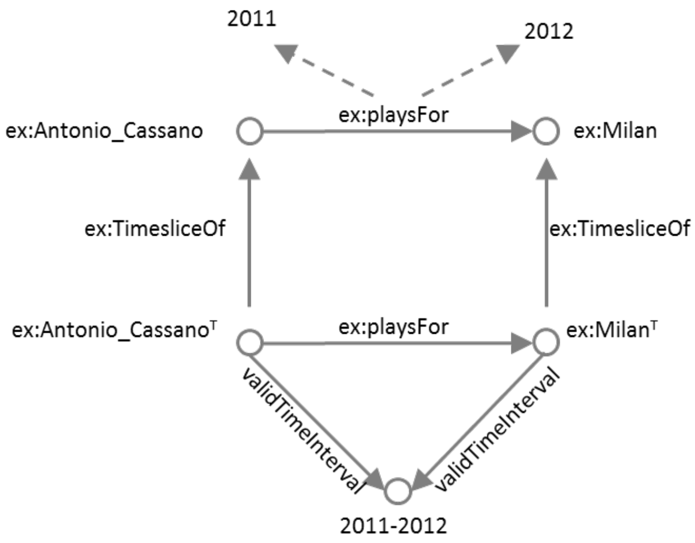


FIGURE 5.6: 4D-fluents-based representation.

Although we could not find any data set adopting this representation approach, well-known ontologies like PROTON[6] and DOLCE[7] adopt it.

[6] http://proton.semanticweb.org/
[7] http://www.loa.istc.cnr.it/DOLCE.html

5.4 Quantitative Analysis on Temporal Information and Temporal Meta-Information

5.4.1 Data set and Experimental Setup

To give more insights about the usage of temporal information in Linked Data cloud, we analyse the latest release of the BTC data set which was crawled from the Web in May/June 2011 using a random sample of URIs from the BTC 2010 data set as seed URIs. The BTC corpus which represents only a part of all available Linked Data on the Web, contains over 2.1 bn triples in N-Quads[8] format with over 47 K unique predicates, collected from 7.4 M RDF documents. Although this corpus does not contain all triples of the LOD cloud, it constitutes a large collection of documents sampled from a wide variety of Linked Data publishers. A crawling-based approach is per design biased towards data sets that are well-interlinked, while more isolated data sets are less likely to be found. We also observe that the corpus is static, and it samples only RDF/XML, not covering data in other syntaxes like RDFa. We expect these aspects not to have any negative effects on the findings of our analysis, which still targets specifically prominent and well interlinked part of the LOD cloud.

Considering the size of the corpus, we use Apache Hadoop[9] to analyse the data. Hadoop allows for the parallel and distributed processing of large data sets across clusters of computers. We run the analysis on the KIT OpenCirrus[10] Hadoop cluster. For our analysis we used 54 work nodes, each with a 2.27 GHz 4-Core CPU and 100GB RAM, a setup which completes a scan over the entire corpus in about 15 minutes.

5.4.2 Temporal Information Analysis

To gather a broad selection of temporal information in BTC, we employ a string-based search method which implements a class named SimpleDateFormat[11] in

[8] http://sw.deri.org/2008/07/n-quads/

[9] http://hadoop.apache.org/

[10] https://opencirrus.org/

[11] http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html

Table 5.1: Top twenty PLDs with respect to temporal quads.

PLD	quad. (M)	Tquad (K)	doc (K)	Tdoc (K)
scinets.org	56.2	3,391	51.9	44.3
legislation.gov.uk	33.1	1,249	246.4	246.4
ontologycentral.com	55.3	1,029	4.6	4.4
bibsonomy.org	34.5	881	234.7	177.3
loc.gov	7.8	854	345.3	302.9
bbc.co.uk	6.3	679	173.5	83.6
livejournal.com	169.8	530	239.2	238.9
rdfize.com	37.6	495	204.7	204.6
data.gov.uk	13.8	479	178.8	91.9
dbpedia.org	28.4	423	596.6	124.1
musicbrainz.org	2.5	359	0.3	0.3
tfri.gov.tw	153.3	272	154.4	78.2
archiplanet.org	16.3	186	79.2	53.5
freebase.com	27.8	173	572.9	109.1
vu.nl	6.8	156	294.2	26.7
fu-berlin.de	5.7	139	291.6	37.4
bio2rdf.org	20.2	129	744.7	71.6
blogspace.com	0.9	124	0.2	0.2
opera.com	24.1	124	160.3	124.1
myexperiment.org	1.5	114	26.1	13.7

Table 5.2: Top twenty temporal properties wrt. temporal quads.

Temporal Property	quad (M)	doc (K)
dcterms:#modified	3.4	44
dcterms:modified	2.3	842
dcterms:date	1.5	247
dc:date	1.4	188
dcterms:created	0.6	450
dcterms:issued	0.2	222
lj:dateCreated	0.2	238
swivt:#creationDate	0.2	197
lj:dateLastUpdated	0.22	225
wiki:Attribute3ANRHP _certification_date	0.18	53
tl:timeline.owl#start	0.17	31
tl:timeline.owl#end	0.15	24
bio:date	0.14	143
po:schedule_date	0.14	15
swrc:ontology#value	0.096	37
cordis:endDate	0.078	0.002
nl:currentLocationDateStart	0.076	26
po:start_of_media_availability	0.074	10
foaf:dateOfBirth	0.068	68
liteco:dateTime	0.062	62

Java. We are confident about the correctness of the collected data because the time parser is well-known and used by a large community of Java developers.

We assume that if temporal information is present, it is contained in the object position of quads. Thus, we use regular expressions to identify temporal information in the object of every quad in the BTC. However, it has been recently shown that the best practices used to publish data on the Web [21] are not always followed by publishers [73].

We notice that often RDF publishers do not use the date formats defined by standards such as RFC 822, ISO 8601 or XML Schema. In order to collect all temporal information that is represented in the BTC but is not fully compliant to standard date formats, we consider variations of the standards. The variations of the standard date formats are expressed by regular expressions based on the following patterns: `(EEE), dd MMM yy (HH:mm:(ss) (Z|z))` and `yyyy-MM-(dd('T'HH:mm:(ss).(s)(Z|z)))` respectively[12]. We extract 12,863,547 *temporal quads*, i.e., quads containing a temporal entity, and 1,670 unique temporal properties from the corpus.

[12] The value in the parentheses is optional.

Furthermore, to provide a deeper analysis of the distribution of temporal information within the data set, we extract all the pay-level domains (PLDs) occurring in the context of the quads. Herein, we use PLDs to distinguish individual data providers [91]. Table 5.1 lists the top 20 PLDs publishing the largest number of temporal quads. For each PLD we report: the total number of quads (*quad.* in Table 5.1), the number of temporal quads (*Tquad.*), the number of documents (*doc*) and the number of temporal documents (*Tdoc*).

We can notice that although `scinets.org` is listed on top of the list, it does not provide the highest ratio of temporal quads over the total number of quads compared to other data sets. With respect to the temporal quads, we can notice that `musicbrainz.org` and `blogspace.com` represent the largest number of temporal quads as a proportion of all quads. Similarly for the documents, we notice that `legislation.gov.uk`, `rdfize.com` and `blogspace.com` represent the three PLDs with the largest number of temporal documents as a proportion of all documents.

Table 5.2 lists the top 20 temporal properties that occur more frequently in the BTC, reporting the number of quads and documents they occur in. We also provide an analysis of the distribution of the top-10 most frequent temporal properties within the most significant PLDs, which is plotted in Figure 5.7. It can be noticed that not only the properties of the Dublin Core (DC) vocabulary[13] do occur much more frequently than other properties, but they are also used more often across different data sets. Remarkably, the temporal property that occurs more often in the BTC data set, i.e., `dcterms:#modified`, has a wrong spelling (the correct spelling denotes in fact the second most frequent temporal property in the corpus). As shown in Figure 5.7, this is also the only temporal property published in the `scinets.org` context, and the spelling is wrong in all the quads having the same context.

5.4.3 Temporal Meta-Information Analysis

In this section we analyse and evaluate the adoption of the approaches for representing temporal meta-information. Our quantitative analysis is augmented by a qualitative discussion in Section 5.6, based on both experiments and literature, to highlight the advantages and shortcomings of each approach.

[13] `http://www.dublincore.org/documents/dces/`

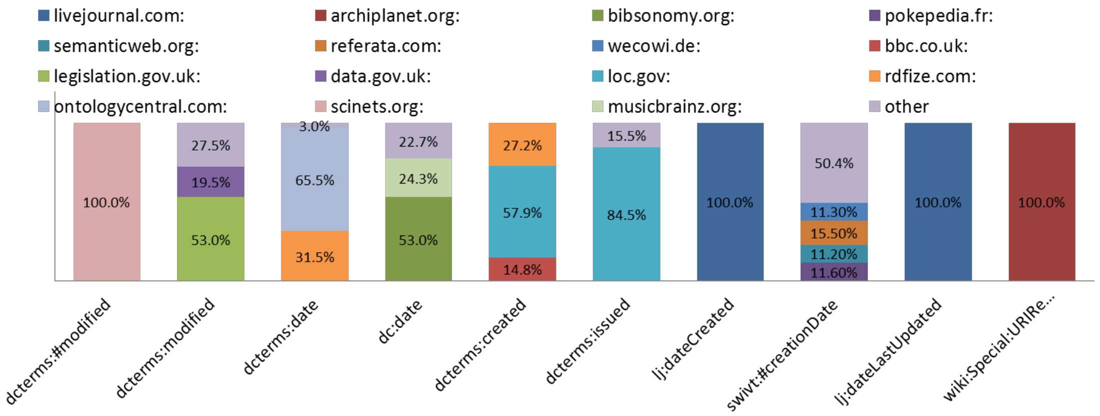


Figure 5.7: Distribution of top ten temporal properties with respect to main PLDs.

Observe that some approaches cannot be at large-scale detected automatically. Therefore, for certain constructs we select a random sample and manually identify the constructs in the sample. We then scale the resulting measure to the entire data set, which consists of 2.1bn quads in 7.4M documents. Of those, 12.8M were temporal quads (containing a date literal) occurring in 2.5M documents.

Analysing larger samples is unfeasible due to the high manual effort involved in checking for constructs in the entire data set; please note that random sampling is an established method for estimating properties of large populations (e.g., the prediction of election outcomes use small samples and achieve sufficient accuracy [10]). For instance, the error bound for Protocol-based representation is +/- 1.9%.

Not all surveyed approaches are adopted on the web. We did not find any uses of the Applied temporal RDF-based representation and the 4D-fluents-based representation in the data. Table 5.3 gives an overview of our findings.

Document-centric approach

To identify the use of the *Protocol-based representation* we ascertain how many of the URIs that identified documents in the BTC return date information in the HTTP header. We generate a random sample of 1000 documents (from the context of the quads), and for each document URI in the sample we perform an HTTP lookup to check the last-modified header in the HTTP response. We found that only 95 out of 1000 URIs returned last-modified headers.

Table 5.3: Temporal meta-information representation approaches and the respective occurrence compared to i) quads having temporal information; ii) overall quads in the BTC; iii) overall documents in the BTC (n/a = not applicable, - = no occurrence).

Granularity	Approach	Occurrence temp. quads	Occurrence overall quads	Occurrence overall docs
Document	Protocol	n/a	n/a	9.5%
	Metadata	5.1%	0.00019%	0.56%
Fact	Reification	0.02%	0.0000008%	0.006%
	Applied temporal RDF	-	-	-
	N-ary relationship	12.24%	0.0005%	0.6%
	4D-fluents	-	-	-

To identify the use of the *Metadata-based representation*, we select a sample of 1000 URIs that appear in the subject position of quads with temporal information. We need to ensure that those subject URIs are in fact documents (information resources), as the Metadata-based representation approach is concerned with documents. Thus, from the sample we exclude URIs containing the # symbol (as URIs with a # per definition do not refer to a document).

For the remaining URIs we send an HTTP request and analyse the response code to determine whether the URI identified a document. We found that 432 (43.2%) identified documents (i.e., directly returned a 200 OK status code). These information resources are not limited to RDF but they also include resources in other formats such as HTML, MP3, XML or PDF. We manually check for RDF documents with only the temporal meta-information such as modified and updated, which resulted in 51 documents.

Of the 51 RDF documents with temporal meta-information in HTTP headers, 43 are also associated with metadata-based dates. Thus, for each of the 43 identified documents we compared protocol-based last-modified and metadata-based last-modified dates. We found that protocol-based last-modified dates are more up-to-date compared to metadata-based dates with an average of almost a year (364 days).

Fact-centric approach

We analyse the *Reification-based representation* in the BTC by looking for how often reified triples contain temporal information. The pattern first identifies the quads containing predicates that are defined in the RDF reification vocabulary (i.e., `rdf:subject`, `rdf:predicate`, and `rdf:object`). From the identified cases we extract only those reified triples that have temporal meta-information associated with their subjects. In the entire BTC data set we found 2,637 reified triples containing temporal meta-information.

To account for *N-ary-relationship-based representation* we again use a combination of sampling of the results of a query over the data set with manual verification since n-ary relations are impossible to identify just by analysing the graph structure. Hence, we sample and manually identify occurrences.

The following pattern identifies for each document triples of the form $\langle s, p, o \rangle$ and $\langle o, p^*, o^* \rangle$ and furthermore identifies whether o is also associated with a temporal entity. Notice that the possibility to join two triples x and y where $x.object = y.subject$ is a necessary, but not sufficient condition, to identify n-ary relations. All results are contained in a set that we name *scoped set* consisting of 7M temporal quads. Hence, from the scoped set, we select three different random samples of 100 triples and we manually verify if respective documents identify an n-ary relation. Results of such manual analysis show that 10, 10 and 12 out of 100 triples in the samples are used with an n-ary relation.

5.5 Recommendations

In the following we discuss the results and provide recommendations for data publishers and consumers.

The approaches that are part of the Document-centric approach are more extensively adopted than the approaches of the Fact-centric approach. As we hypothesised, the number of temporal meta-information associated with documents is greater than those associated with facts. Still, the use of temporal meta-information for documents (about 10% overall) are not sufficiently high enough to support our outlined use case.

We identify two approaches used for annotating documents with temporal meta-information: the ***Protocol-based representation*** and the ***Metadata-based representation***. We notice that the number of temporal meta-information are much more available in the Protocol-based rather than the Metadata-based representation. The temporal meta-information in the HTTP header, when available, are more up-to-date than the ones in the RDF document itself. *Consumers:* The applications that consume temporal meta-information should first check for temporal meta-information in the Protocol-based representation because they are more up-to-date; in case this information is not available the applications should be able to check in the Metadata-based representation. *Publishers:* Publishers should carefully update the temporal meta-information whenever the data in the document is changed; temporal meta-information in both Protocol- and Metadata-based representation should be consistent.

We identify four approaches used for annotating facts with temporal meta-information, grouped into the Sentence-centric approach and the Relationship-centric approach. These approaches associate validity expressed as temporal entities to facts.

The use of the ***Reification-based representation*** show a high complexity w.r.t. query processing [75]. The approach appears only in a very small number of quads. *Consumers:* Consumers should be able to evaluate based on the application scenario (e.g., the expected types of queries) if it is possible to either build their applications over such representation or to choose a different, and more efficient approach (e.g. Applied temporal RDF-based representation). *Publishers:* Publishers should be aware that best practices discourage the use of Reification-based representations, as they are cumbersome to use in SPARQL queries [21], even though they may be useful for representing temporal meta-information.

The performance of ***Applied temporal RDF-based representation*** has been reported to have still some efficiency issues [131], especially in the worst case, when the number of graphs (which are associated with temporal entities) is almost equivalent to the number of triples. *Consumers:* Although we found no usage of the Applied temporal RDF-based representation in the BTC, the approach should deserve more attention because it supports expressive temporal queries based on τ-SPARQL, and can be applied to data sets that provide temporal information according to a Reification-based representation. *Publishers:* Publishers should take into consideration the worst case when using the Applied temporal RDF-based representation. Therefore, they should use it only when it is possible to group a considerable number of triples into a single graph.

The ***N-ary-relationship-based representation*** embeds time in an object that represents a relation. In the BTC, 0.6% of documents contain at least one case of N-ary-relationship-based representation, which is greater than the Reification-based representation but still represents only a small fraction of the overall number of documents. *Consumers:* Consumer applications can evaluate the temporal validity of facts from representations based on this approach. The lack of a clear distinction between plain temporal information and temporal meta-information provides high flexibility, but at the same makes difficult to predict the kind of temporal information that can be leveraged and interpret its meaning. Collecting these temporal meta-information with automatic methods is not straightforward, as shown by the manual efforts required in our analysis to identify this information. *Publishers:* Many situations require temporal meta-information associated with relations that can be modelled only as complex objects. Therefore, we recommend to publishers to use N-ary-relationship-based representation for complex modelling tasks because it allows flexibility on representing temporal meta-information associated with relation.

The ***4D-fluents-based representation*** supports advanced reasoning functionalities, but, probably also because of its complexity, has not been adopted on the Web.

5.6 Summary

The key contribution of this chapter is the identification fo different approaches used to represent temporal information in Linked Data on the Web, which is important for several research and application domains. As time introduces a further dimension to data it cannot be easily represented in RDF, a language based on binary relations; as a result, several approaches for representing temporal information have been proposed. Based on the qualitative and quantitative analysis using the Billion Triple Challenge 2011 dataset, we came to the conclusion that the availability of temporal information describing the history and the temporal validity of statements and graphs is still very limited. If the representation of temporal validity of RDF data is somewhat more complex and can be expected to be considered in specific contexts, information about the creation and modification of data can be published with quite simple mechanisms. Yet, this information would have great value, e.g., when data coming from different sources need to be integrated and fused.

Chapter 6

Data Freshness in Linked Data

In this chapter we address the problem of defining measures for assessing data freshness (also known as data currency measures Section 4.1.5) to support the consumer to evaluate how fresh the explored or queried data is. We aim to provide a method for measuring data freshness at two levels of granularity, that is, the freshness at triple and graph level (e.g., the graph returned by a query).

The chapter is organized as follows: Section 6.1 discusses why knowledge about data freshness is important to LD applications and consumers; the latter case is further motivated by a practical example. Due to the diversity of temporal annotation properties used in LD, Section 6.2 presents a conceptual model, which aggregates temporal annotation properties used in different vocabularies to support the design of an interoperable technique for data freshness assessment. In Section 6.3 we describe a framework for assessing data freshness of RDF documents when temporal annotations of documents are available. In Section 6.4, we propose an alternative framework for data freshness assessment when temporal annotations of documents are incomplete and inaccurate, based on the estimation of last modification date and then describe the framework for assessing data freshness of DBpedia entity documents. We then introduce in Section 6.5 two measures for data freshness of RDF data elements. We show results of our experimentation on DBpedia data set in Section 6.6. Finally, Section 6.7 provides a summary of the chapter.

6.1 Overview

Information providers need to be aware about the freshness of the data sets and deal with the dynamic nature of LD [130], before integrating and presenting them to the users. On the other hand, the freshness of a data set is important for users and applications that consume data. Data freshness plays an important role in the success of the information systems [141, 99] and is also considered as one of the most important dimensions of data quality for data consumers [127]

The problem of data freshness arises in various application domains such as:

- *Data integration.* Linked Data technology is designed mainly to unify same real-world objects coming from different sources in a single data space. To fulfill this task, multiple conflicting values of the same attribute need to be fused. An important role in resolving these conflicts is played by data freshness which prefers values with high freshness.
- *Querying.* The execution of SPARQL queries over a large number of Linked Data sources, like a distributed database system, provides enormous potential [64]. The advantage of sending a query over different data sets and not only on a centralized repository, is that, the query result returned is more likely to be current than the one returned from the centralized repository where the data need to be updated and maintained. A distributed system benefits of having the same information distributed in several data sets (e.g. SQUIN[1]).
- *Caching Web Data.* The triples are saved locally in order to improve query efficiency in the Web of Data. A caching system in order to be current should applies replacement policies which allows the system to consider the most updated data and replace out-of-date data.
- *Data warehouse.* A data warehouse maintains a local snapshot, called a materialized view, of underlying data sources, This materialized view should be updated in order to reflect the actual sources state, so the goal is to assure a certain goal of data freshness.

[1] http://squin.sourceforge.net/index.shtml

- Semantic Web Search Engines. A semantic web search engine [74] (e.g SWSE[2]) provides access to billions of triples with search results semantically aggregated from multiple documents. The ranking of *elements* such as entities, triples, sources, etc., is provided through TR-IDF relevance scores. Consider the integration of the freshness concept into link-analysis algorithms. Along this line, the ranking will be based not only on the importance of the elements but will additionally reflect the freshness of the elements to search users. To achieve this, the search engines would have to accurately capture temporal annotation in both the last update of the elements as well as the last update of the links which are specified simply by using external URI names in the data.

Motivation Example

Usually, temporal meta-information when available more often annotate documents rather that triples. We intuitively suggest that the assessment of triples freshness can provide an insight into the question whether a fact is still semantically accurate at a given time (Intuitively an entity document can be considered semantically accurate when it accurately represents a real-world entity (see Chapter 4)). Observe that, the more up-to-date data is, the more the user can be confident that data is still semantically accurate [97]).

Consider a consumer who need to assess the freshness of an answer graph to guarantee that the set of triples contained in the answer is semantically accurate at the time the query is submitted. In particular, the following SPARQL query asks for the list of soccer players currently playing for the national team *A.C Milan*, the club number, the number of goals and the number of appearances.

```
SELECT distinct ?team ?playerUri ?playerName WHERE
 {
  ?playerUri foaf:name ?playerName .
  ?team rdf:type dbo:SoccerClub .
  ?team rdfs:label "A.C. Milan"@en .
  ?playerUri dbpedia2:currentclub ?team .
  ?playerUri dbpedia2:clubnumber ?number .
  ?playerUri dbpedia2:goals ?goals .
  ?playerUri dbpedia2:caps ?caps .
```

[2] http://swse.org/

```
    }
  ORDER BY(?name);
```

In Figure 6.1 we consider the case of two popular soccer players which can be generalized to the domain of all soccer players. We compare the facts returned by DBpedia with the facts in the original source, that is Wikipedia. Lionel Messi's infobox provides different information compared to the DBpedia entity documents for goals and appearance values. Instead, Cristiano Ronaldo's DBpedia entity document provides different values available on Wikipedia for all properties. As one can notice, the example shows that there exist a relation between last modification date and the semantic accuracy of the information observed in March 3ed 2014.

This example, highlights one more time the need to study data freshness of the answer graph or the triples contained in the answer graph. The result of a query is obtained using an answer graph, which is an unnamed graph. The problem is to assess the freshness of unnamed graphs. If we have a method to assess the freshness of an unnamed graph, we can assess the freshness of an answer graph to a given query. Although data freshness has been largely studied in other areas such as relational databases [12] or data integration system [25], several challenges exists to data freshness assessment in LD.

Vocabulary heterogeneity - Temporal annotation properties available from the LD sources are represented using different vocabularies and ontology properties due to data set diversity and autonomous providers. For example, two properties *uniprot:modified* and *drugbank:updated* represents both the modification date but belongs to different vocabularies respectively to UniProt[3] and DrugBank[4]. The semantic mismatches among these properties need to be solved in order to avoid the design of many source-specific algorithms for data freshness assessment.

Availability of temporal meta-information of documents - Temporal meta-information when available more often annotate documents rather than triples or answer graphs. Answer graphs denoted as G^A (see Section 2.1) are typically unnamed graphs. In contrast, documents contain usually named graphs

[3] http://www.uniprot.org/downloads

[4] http://www4.wiwiss.fu-berlin.de/drugbank/

Table 6.1: Dynamic properties comparison of DBpedia entities with different freshness

Entity	http://dbpedia.org/resource/Lionel_Messi			http://dbpedia.org/resource/Cristiano_Ronaldo		
Property	Wikipedia	DBpedia	Matching Values	Wikipedia	DBpedia	Matched Values
Current club	A.C. Milan	A.C. Milan	1/1	A.C. Milan	Real Madrid C.F.	0/1
Club number	11	11	1/1	22	8	0/1
Goals	{5, 6, 230 }	{5, 6, 212 }	2/3	{3, 84, 169 }	{3, 84, 140 }	2/3
Appearances	{10, 22 ,265 }	{10, 22, 243}	2/3	{25, 196, 157 }	{25, 196, 129 }	2/3
Last-Modified	02/Mar/2014	01/Sep/2013	6/8	02/Mar/2014	08/May/2013	4/8

and since these graphs have a name they can be easily annotated. The size of data sets is usually too big to provide temporal meta-information annotating triples or unnamed graphs.

Incompleteness and inaccuracy of temporal meta-information - The assessment of data freshness would be trivial if all data elements carried temporal annotations and these information were represented as temporal meta-information in a data set (see Section refsec:concept). In practice, as we showed in Chapter 5, the temporal meta-information is very limited. In addition, temporal meta-information when available, can be incomplete and inaccurate.

The main goal of this chapter is to study approaches for measuring freshness of RDF data elements. As explained in Chapter 4, data freshness assessment needs versioning metadata. The first proposed method studies how to assess freshness of unnamed graphs and individual triples when temporal meta-information annotate documents rather than individual triples or unnamed graphs. While in the first method we assume that there exist available temporal entities annotating documents, in the second method we study how to assess freshness of all RDF data elements when both temporal meta-information annotating documents and triples are incomplete and inaccurate.

6.2 Vocabulary Heterogeneity

The temporal annotation properties are not semantically related by mappings, which means that it is not possible for a machine to understand that, e.g., the `dcterms:modified` and `ecowlim:dateCollected` properties bear semantically comparable information. Thus, with the aim to allow an automatic machine discovery process, we built an ontology including the definition of each temporal annotation property and also the mappings that we defined between them. We provide a sharable and interoperable conceptual model, named OntoCurrency, which aggregates heterogeneous temporal annotation properties related to the assessment of data freshness, according to an alignment and mapping across different data sets. To realize the OntoCurrency, we use the temporal annotation properties extracted from the BTC data set (Chapter 5).

OntoCurrency[5] is an ontology represented in the OWL 2[6] language, which integrates in one place all temporal annotation properties. The ontology therefore collects the temporal annotation properties that can be used for data freshness measurement and describes their mutual relationships, thereby enabling different applications to compute the freshness of graphs and facts represented in different data sets. The prefix we use for our ontology is *oc:*. As depicted in Figure 6.1, we distinguish among three core datatype properties (all these properties are subproperty of `owl:topDataProperty`): `oc:modification`, `oc:creation` and `oc:lastModification` (see Figure 6.1).

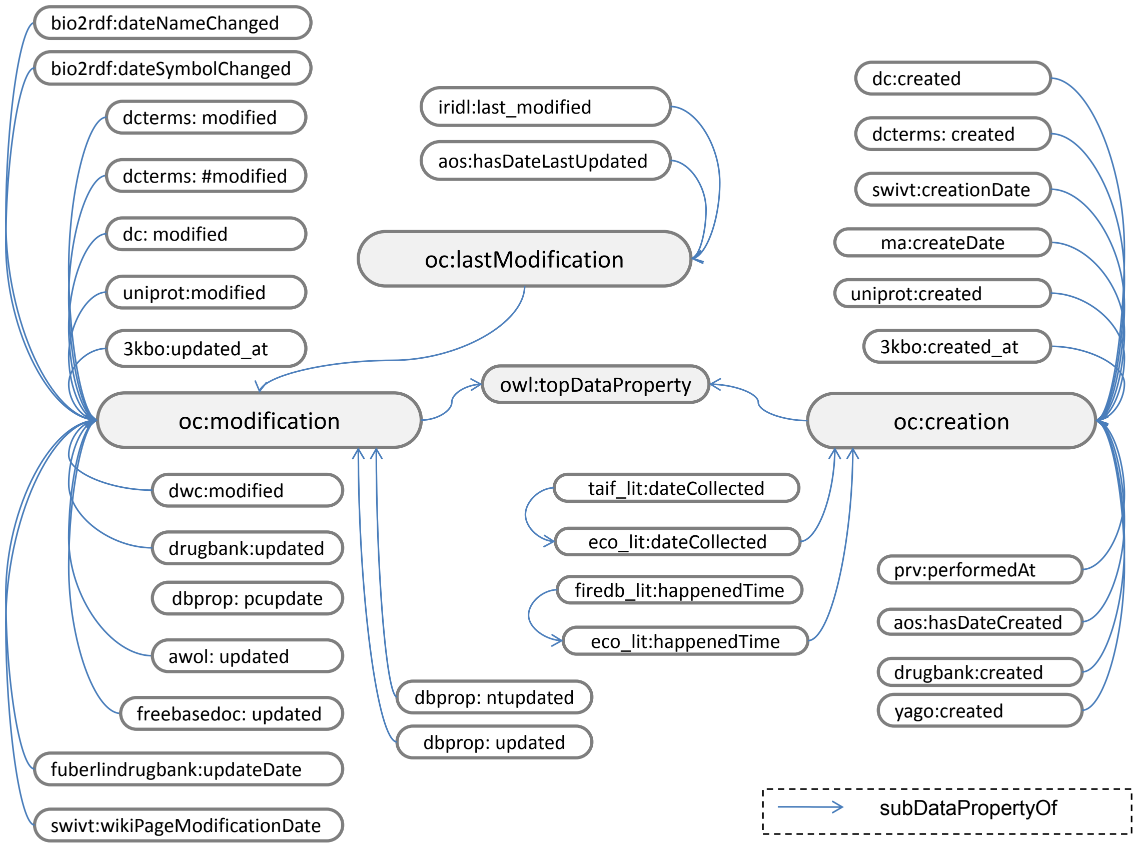


FIGURE 6.1: The OntoCurrency ontology

The property `oc:modification`, and all the properties defined as subproperty to the core property, specify the date on which an entity document has been modified. The subproperty relations between the local property defined in our ontology and other properties having the same meaning facilitate application interoperability. If a reasoner is available, triples representing information about the modification of

[5] http://siti-server01.siti.disco.unimib.it/research/experimental-results-and-prototypes/ontologies/OntoCurrency.owl

[6] http://www.w3.org/TR/owl2-overview/

an entity document that use different properties can be retrieved by querying each of the subproperties. The properties used to describe information about the last modification of an entity document are considered subproperties of the core property modification. To do so, we introduce a local property `oc:lastModification` as subproperty of `oc:modification`. As a consequence, one can retrieve metadata about the last modification of an entity document (e.g. including the properties `aos:hasDateLastUpdated`) when he looks for generic metadata about its modification. The last property introduced in our ontology is `oc:creation`, which specifies the creation date of an entity document. We will show that this can be used to assess freshness when the temporal annotation property about a modification date is not available.

6.3 Availability of Temporal Meta-information of Documents

OntoCurrency is used to support the design of an interoperable technique for data freshness assessment by leveraging the different vocabularies used in most of data sets in LD. In the following we propose a general purpose approach for assessing data freshness of a single entity document:

1. INPUT: An entity e represented by an HTTP URI.
2. Access the entity document d^e given the entity e by:
   (a) dereferencing e by looking up the HTTP URI based on the HTTP protocol, or
   (b) using a SPARQL query language with the IRI provided from the entity e in the GRAPH clause set to match patterns against the named graph, or
   (c) downloading the RDF dump of the document entity d^e.
3. Retrieve the temporal meta-information available for d^e by:
   (a) performing a lookup in the HTTP header returned in response to an HTTP GET request for the document d^e and retrieve the Last-Modified value (see Protocol-based representation in Chapter 5), or

(b) identifying temporal meta-information annotating RDF document d^e by considering the temporal annotation properties that belong to the OntoCurrency ontology (see Metadata-based representation in Chapter 5).

4. OUTPUT: Data freshness of d^e

We now explain in details steps 3 of the approach, while step 4 will be explained in detail in Section 6.5.1 and Section 6.5.2.

Let P^M and P^C be two sets of temporal annotation properties expressing modification and creation, respectively and which are defined as follows: $P^M = \{a|a \sqsubseteq oc{:}modification\}$ and $P^C = \{a|a \sqsubseteq oc{:}creation\}$. We now define the *modification annotation graph* of a document d^e as the set of triples $G^M(d^e) = \{\langle s,p,o\rangle | s = e^s \wedge a \in P^M\} \cup triplify(LastModHTTP(d^e))$; the graph represents all the triples, where e^s is the subject and $a \in P$ is a property from the set P^M and $triplify(LastModHTTP(d^e))$ is a function that creates a triple with subject e and property $oc{:}lastModification$ from the last modification date of the document represented in the document's HTTP header (if this temporal entity is available). Analogously, the *creation annotation graph* of a document d^e is the set of triples $G^C(d^e) = \{\langle s,p,o\rangle | s = e^s \wedge a \in P^C\}$; this graph represents all the triples, where e^s is the subject and $a \in P$ is a property from the set P^C.

6.4 Incompleteness and inaccuracy of temporal meta-information

We provide an assessment framework that leverages the temporal entities extracted from Wikipedia to compute the data freshness of DBpedia entity documents; the assessment framework follows three basic steps sketched in Figure 6.2: (1) extract a document representing an entity from the DBpedia data set, (2) estimate the last modification date of the document looking at the version history of the page that describes the entity in Wikipedia, (3) use the estimated date to compute data freshness values for the entity document.

To assess the data freshness of RDF documents, we propose an assessment strategy based on the Age-Based and System currency metric. Our assessment framework takes a DBpedia entity as input and returns a Age-Based currency value

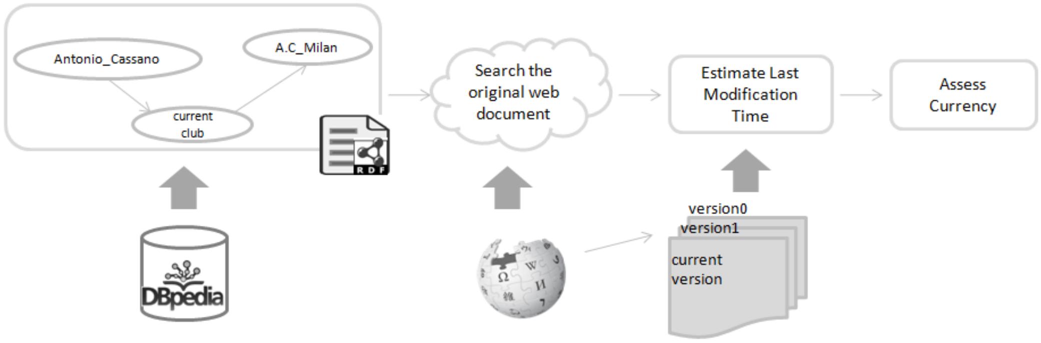


FIGURE 6.2: Main components of the framework for assessing data freshness of DBpedia entities

and a System currency value. We propose a method, similar to the data extraction framework proposed in [112], which uses versioning metadata available in Wikipedia pages which are associated with time-stamped global version identifiers, to extract the time-stamps to associate to RDF documents as last modification date.

```
Algorithm 1: DBpedia Data Freshness
    Input: An entity $e$
    Output: Data freshness values
    $result = \phi$
    $d^e = rdfFetcher(e)$
    $v = getLastVersionId(p^e)$
    $\overline{d}_v^e = buildRdf(v)$
    while $\overline{d}_v^e$ is not equal to $d^e$ do
        $v = getPreviousVersionId(p, v)$
        $\overline{d}_v^e = buildRdf(p, v)$
    $\tau = getTimestamp(v)$
    $\beta(d^e) = computeAgeBasedCurrency(\tau)$
    $\sigma(d^e) = computeSystemCurrency(\tau)$
    return $Datafreshnessofd^e$
```

A pseudocode of the algorithm that implements the strategy is described in Algorithm 1. The input of the algorithm is an entity e for which the data freshness has to be evaluated using the data freshness measures. To obtain the estimated last modification date of an entity document we need to extract its entity document and its correspondent page from the web.

The entity document d^e is obtained by the function $rdfFetcher(e)$, line 2, which reads the triples recorded in an N-Triples entity document and records them. Among all the triples in the document we keep only those that use DBpedia properties; considered documents represent relational facts and do not include typing information, links to other data sets (e.g., same as links and other links to categories); in other words we consider in entity documents statements that are more sensitive to changes.

From line 3 to line 4, the algorithm extracts the ID of the last revisioned web page p^e corresponding to the entity e and builds a structured representation of p^e. In order to build a structured RDF content we need to identify properties and their values in the semi-structured part of the document (p^e). The algorithm creates RDF triples from p^e by using the same approach and mappings used to extract DBpedia triples from Wikipedia infoboxes [22].

Given p and d^e, the algorithm finds whether the structured representation of p provided by $\overline{d}_v^e$ matches the entity document d^e (see line 4-7); we use an exact match function. In case the last revision of the structured representation of p does not match to the entity document, the algorithm checks for older versions and stops only when a matching version is found. At this point, we associate the timestamp of the v version to the entity document d^e (line 8). In this way we compute the data freshness of d^e (see line 9-11). Data freshness formula will be explained in detail in the next section where we provide two measures for data freshness named *Age-Based currency* (Section 6.5.1) and *System currency* (Section 6.5.2).

6.5 Measures for Data Freshness

This section describes two measures of data freshness of RDF data elements. The first measure based on the age of data elements, named Age-Based currency, propagates data freshness of answer graphs or individual triples based on the data freshness of the entity documents that describe the entities occurring in the answer graph. The second measure based on the delay between the LD source and the original one, named System currency, considers that the original source has a versioning mechanism where each version is identified by a time point. We start by defining the problem we solve in each section and then provide Age-Based currency (Section 6.5.1) and System currency (Section 6.5.2) respectively.

6.5.1 Measure based on the age of data elements

Problem Statement. In this section, we consider the following problem: given (1) a set of facts F returned as a result of querying a data set or several data sets; (2) for each fact $\langle s, p, o\rangle \in F$, with subject identified by the entity e^s and object identified by the entity e^o; and (3) the date literal triples $\langle d^{e^s}, a^s, t^s\rangle$ and $\langle d^{e^o}, a^o, t^o\rangle$ each representing the entity document of the subject and the object respectively, the temporal annotation property and the time points t^s and t^o associated with the entity document of the subject and the object respectively; we would like to estimate a freshness score for each fact as $(\langle s, p, o\rangle, datafreshness) \in G^A$.

The freshness assessment of the fact problem propagates the freshness of the entities appearing in the triple to the freshness of the fact itself. Therefore, if we have a method to associate an entity with a freshness score, we are able to associate freshness at the fact level containing that entity and further provide the freshness of the G^A as a whole.

In the following section, we address the above problem, by providing a first formula to measure freshness in LD, based on the age of the data elements, and a method to assess the freshness of an RDF graph/triple, based on the freshness of the entity documents that describe the entities occurring in the graphs (triple).

Age-Based currency

As explained at the beginning of this chapter data freshness is captured by data currency dimension. According to the literature for relational databases 4 we now define a metric for measuring data freshness named *Age-Based currency* based on the time of last modification of an entity or a fact.

Definition 6.1 (Age-Based currency)**.** *The currency of a data element is defined as the age of the data element*, where the age of the data element is computed as the difference between the current time (the observation time) and the time when the data element is last modified.

Let D be a set of entity documents describing a set of entities E, and $cTime$ and $lmTime(d^e)$ be respectively the *current time* and the last modification time of an entity document d^e. We first define the *age measure of an entity document*, based on the above informal definition of data freshness measure. The $age(d^e) : D \rightarrow [0 \cdots + \infty]$ of an entity document d^e, with $e \in E$, can be measured by considering the time of the last modification of the entity document according to the following formula:

$$age(d^e) = cTime - lmTime(d^e) \tag{6.1}$$

Age-Based currency depends on the document *age* and is a normalized value in the range $[0, 1]$. Freshness values are high when the document are up-to-date and low when documents are old. The Age-Based currency $\beta(d^e) : D \rightarrow [0, 1]$ of an entity document d^e is defined as follows:

$$\beta(d^e) = 1 - \frac{age(d^e)}{cTime - startTime} \tag{6.2}$$

where $startTime$ represents a time point from which data freshness is measured (as an example, $startTime$ can identify the time when first data have been published in the LD). Observe that this Age-Based currency measure does not depend on the particular nature of RDF document. In fact, the same measure can be adopted to evaluate data freshness of other web documents such as Wikipedia pages.

Data freshness assessment of answer graphs

The last step of the framework proposed in Section 6.3 can be split in two steps. A first step includes the assessment of age measure of entity documents, which can by realized by applying Equation 6.1, where $age(d^e)$ is defined as follows:

$$age(d^e) = \begin{cases} cTime - lmTime(d^e) & \text{if } G^M(d^e) \neq \emptyset \\ cTime - lcTime(d^e) & \text{elsewhere} \end{cases} \tag{6.3}$$

The notation $lcTime(d^e)$ is used to identify the creation time of an entity document d^e. The values of $lmTime(d^e)$ and $lcTime(d^e)$ are calculated by considering the most recent (maximum) among the values available. The values of $lmTime(d^e)$ and $lcTime(d^e)$ extracted from LD sources often come in different data formats. To solve this heterogeneity we use a date parser provided as a class in Java known as *SimpleDateFormat*[7], which also provide the normalization of heterogeneous data format into a common format. The second step includes the assessment of Aged-based currency given by Equation 6.2.

Now that we are able to assess freshness of an individual entity document, we can compute the data freshness of a set of entity documents, and hence, of the answer graphs. The data freshness of a set of entities can be defined as the average freshness of the entity documents in the set. Let $E = \{e_1, ..., e_n\}$ be a set of entities; the freshness of E is defined as:

$$cur(E) = AVG(\beta(d^{e_i})) \tag{6.4}$$

with $1 \leq i \leq n$.

The above formula can be adopted to evaluate the freshness of a graph G, by considering the set of entities occurring in a graph, i.e., by setting $cur(G) = cur(E(G))$; the freshness of a triple can be defined in an analogous way. However, sometimes it is not possible to evaluate the freshness of all entities occurring in a graph, because some documents describing the entities can have no temporal meta-information available. We therefore define the *data freshness completeness* metric (dfc) to evaluate the completeness of the data freshness measurement, measured on a per triple basis. Given a graph G, data freshness completeness evaluates the number of entities for which data freshness can be computed over the total number of entities occurring in a graph. Let $E^+(G)$ be the set of entities occurring in a graph G such that have a data freshness value different from zero; $cc(G)$ can be defined as follows:

$$dfc(G) = \frac{|E^+(G)|}{|E(G)|} \tag{6.5}$$

6.5.2 Measure based on the delay between two sources

The main problem we have to face, in order to apply the data freshness model described in Section 6.5.1, concerns the unavailability of the temporal entities associated with documents required to compute the data freshness of data elements. According to Chapter 5, the empirical analysis shows that only 10% of RDF documents are estimated to be associated with temporal entities. It is challenging to estimate the last modification value.

[7] http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html

A possible solution is by adopting a revision control practice where each change is identified as a revision with a temporal entity. The work in [37] proposes a resource versioning mechanism for Linked Data which allows to publish time-series of entity documents changing over time; however this mechanism has been adopted only by a limited number of data sets.

Problem Statement. Let $E = \{e_1, e_2, \ldots, e_n\}$ be a set of entities and $D = \{d^{e_1}, d^{e_2}, \ldots, d^{e_n}\}$ be a set of entity documents. Let S be a non structured or semi-structured web source that consists of a set of web pages $S = \{w_1, w_2, \ldots, w_n\}$ we assume that each page commonly adopt revision control practice where each change is identified as a version with a temporal entity. We assume that each entity e_i represented by an entity document d^{e_i} has a corresponding web page $p_i \in S$; in the following we use the notation w^e to refer to the web page that describes the entity e. Entity documents and source web pages change over time. Web pages adopted with the revision control mechanism are able to manage changes of information. Formally, a version of a web page w can be represented as a triple $v = \langle id, w, t\rangle$, where id is the version identifier, w is the target web page, and t is a time point that represents the time the version has been published.

In this section, we formalise the problem as follows: given 1) a triple $\langle s, p, o\rangle$ with subject identified by the entity e^s, 2) given the entity document d^{e^s} of the entity e^s containing the triple $\langle s, p, o\rangle$, and 3) the version of the web page w^{e^s}, $v = \langle id, w^{e^s}, t\rangle$; we would like to estimate the last modification date of the entity document as $(d^{e_i}, lmTime(d^{e_i}))$.

Further, as we saw in Section 6.5.1, Age-Based Data freshness is based on age and strongly depends on the observation time. We introduce another measure, which is less sensitive to observation time (observation time is used only for normalizing the values returned by the measure).

In the following section, we address the above problem, by providing a second formula to measure freshness in LD, based on the delay with which data are extracted from an original web source enabled with a versioning mechanism.

System currency

Inspired by the definition of data freshness (also named currency) proposed for relational databases, we adopt and provide a new one for LD.

Definition 6.2 (System currency)**.** *Currency refers to the speed with which the information system state is updated after the real-world system changes* [139].

According to this definition data freshness measures the temporal delay between changes in the real world and the consequent updates in the data. An *ideal* system of currency measure in our domain should evaluate the time elapsed between a change affecting a real-world entity and the update of the RDF document that describes that entity. However, changes in the real-world are difficult to track because real-world is opaque to a formal analysis. Since the data under the scope of our investigation are extracted from a web source, we can define the System currency of an RDF document by looking at the time elapsed between the update of the web source describing an entity and the update of the correspondent RDF document.

We first define the notion of system delay with respect to an entity, defined by a function $systemDelay(d^e) : D \rightarrow [0 \cdots +\infty]$ as the difference between the time of last modification of a web page p^e and the time of last modification of its respective entity document d^e as follows:

$$systemDelay(d^e) = lmTime(p^e) - lmTime(d^e) \tag{6.6}$$

Based on the measure of system delay, we introduce a new data freshness measure called *System currency* that returns normalized values in the interval $[0, 1]$, which are higher when the data are more up-to-date and lower values when data are less-up-to-date with respect to the web source. The System currency $\sigma(d^e) : E \rightarrow [0, 1]$ of an entity document d^e is defined as:

$$\sigma(d^e) = 1 - \frac{systemDelay(d^e)}{cTime - startTime} \tag{6.7}$$

6.6 Experimental Evaluation

To evaluate the framework, we perform a complete experimentation on DBpedia and we also evaluate the quality of proposed data freshness dimensions by means of two metrics we propose in this chapter.

Evaluation of data freshness

When multiple metrics are defined the problem of evaluating the quality of such dimensions arises. In this section we propose a method to evaluate the effectiveness of data freshness measures addressing entity documents extracted from semi-structured information available in web sources. One of the key reasons for computing data freshness of entity documents is that out-of-date documents can contain facts that are not semantically accurate anymore when data is consumed. A data freshness measure that is highly correlated to semantic accuracy is more useful than another measure that is weakly correlated to semantic accuracy, because the former can provide a user with insight into the reliability of the data she/he is consuming. However, the definition of semantic accuracy as correspondence to real-world (even if mediated by a judgment from an expert) is opaque to a formal analysis and difficult to use in practice. Assuming that the source is reasonably reliable (recent studies provide evidence that, e.g., Wikipedia is reasonably reliable even in controversial domains such as Socio-Politics [29]), we define two metrics, namely *accuracy* and *completeness*, to capture the intuitive notion of semantic accuracy by comparison with the web source which data are extracted from.

These metrics use the semi-structured content available in the web source - Wikipedia infoboxes in our case - as a Gold Standard against which entity documents are compared.

We define a *Wikipedia fact* as a triple $\langle c, ip, iv \rangle$ where c represents a wikipedia concept, ip represents an infobox property, and iv represents the value associated with c by the property ip in the infobox. Let us assume to have a mapping function μ that map every Wikipedia fact to an RDF fact. This mapping function can be the one used to extract DBpedia facts from Wikipedia facts[8]; however other techniques to extract RDF facts from semi-structured content have been proposed

[8] http://mappings.dbpedia.org

[115]. A DBpedia fact s is semantically accurate at a time point t iff there exist a Wikipedia fact w in a page version $v = \langle id, p, t' \rangle$ such that $\mu(w) = s$, with $t' \leq t$ and v being the last page version at t.

We define the accuracy $A(d^e)$ of an RDF document d^e as the number of semantically accurate facts $VF(d^e)$ in d^e divided by the total number of facts $TF(d^e)$ in d^e:

$$A(d^e) = \frac{VF(d^e)}{TF(d^e)} \tag{6.8}$$

We define the completeness of an RDF document d^e as the number of semantically accurate facts $VF(d^e)$ in d^e divided by the total number of existing facts $WF(p^e)$ in the original document p^e:

$$C(d^e) = \frac{VF(d^e)}{WF(p^e)} \tag{6.9}$$

Experiment setup

In order to produce an exhaustive and significant experimentation, we define a specific subset of DBpedia entities that is a representative sample for the entire data set. The class of DBpedia entities we consider here belong to the *Soccer Player* class. The main facts describing a soccer player in the Wikipedia's infobox template are shown in Listing 6.1. In particular, the facts of a soccer player such as his appearance, his goals or moving from one club to another, denote evidences about the changes performed. The changes can usually vary from three to seven days which implies a high frequency of modifications. Furthermore, after observing for several months the modification set of the Wikipedia's infoboxes of the soccer players, we noticed that there is a high interest of the end-users to maintain the infoboxes up-to-date. Often due to the high frequency of changes in Wikipedia, the new modifications are not replicated also to DBpedia as shown from the example in Table 6.2.

Listing 6.1: Infobox of Mario Balotelli soccer player

```
{{Infobox football biography
| name                = Mario Balotelli
| birth_place         = [[Palermo]], Italy
| currentclub         = [[A.C. Milan|Milan]]
| clubnumber          = 45
| years1             = 2005-2006
| clubs1 = [[A.C. Lumezzane|Lumezzane]]
| caps1 = 2   | goals1 = 0
| years2             = 2006-2010
| clubs2 = [[Inter Milan|Internazionale]]
| caps2 = 59  | goals2 = 20
| years3             = 2010-2013
| clubs3 = [[Manchester City F.C.|
           Manchester City]]
| caps3 = 54  | goals3 = 20
| years4             = 2013-
| clubs4 = [[A.C. Milan|Milan]]
| caps4 = 3   | goals4 = 4
...
| club-update          = 15 February 2013
| nationalteam-update = 6 February 2013
}}
```

In the example, we provide a comparison between properties that tend to have a high frequency of modification for two popular soccer players.

Mario Balotelli's infobox provides different information compared to the DBpedia entity document. Instead, Giampaolo Pazzini's DBpedia entity document provides the same information available on Wikipedia. This example shows that there is a relation between the update of these sensitive properties and the data freshness measures computed on 15th of February, 2013.

Comparison of the two approaches

To validate the effectiveness of the data freshness framework proposed in this work, we compare the Age-Based currency (Section 6.5.1) for the assessment of entity documents data freshness according to the approaches proposed in Section 6.3 and Section 6.4. For simplicity we name *A-TMI*, the approach where temporal meta-information annotating documents is available but not complete or accurate and *VM-TMI*, the approach where temporal meta-information for annotating documents are not available but are made available through the versioning mechanism of the original source. As we can see from Table 6.3, it is possible to notice that our approach performs better than the previous approach with respect to the number of entities that are assessed. While the technique proposed in this section assesses

Table 6.2: Dynamic properties comparison of DBpedia entities with different data freshness values

Entity	http://dbpedia.org/resource/Mario_Balotelli			http://dbpedia.org/resource/Giampaolo_Pazzini		
Property	Wikipedia	DBpedia	Matching Values	Wikipedia	DBpedia	Matched Values
Current club	A.C. Milan	Manchester City F.C.	0/1	A.C. Milan	A.C. Milan	1/1
Club number	45	45	1/1	11	11	1/1
Goals	{0, 20, 20, 4}	{0, 19, 20}	2/4	{12, 25, 36, 16, 10}	{12, 25, 36, 16, 10}	5/5
Appearances	{2, 59 ,54 ,3 }	{2, 59, 40}	2/4	{51,108, 75, 50, 20}	{51,108, 75, 50, 20}	5/5
$\beta_{DBpedia}$	0.8517			0.9731		
σ	0.6371			1		

all the entities, the previous approach assess only 13% of them. The low number of the assessed entities in the second approach is because either the triples expressing the last modification date of the document are missing or they contain wrong date format. We also distinguish between the 87% of the non assessed entities where 68% of them represent triples with wrong date format (e.g., 20 xsd:integer) and 32% of them does not contain a temporal meta-information associated with the document.

Table 6.3: Comparison of the effectiveness between the approaches

Model	#Entities	#Entities assessed
A-TMI	97	13
VM-TMI	97	97

The comparison among approaches presented in this chapter for the 13 entities of Table 6.3 are shown in Figure 6.3. The first two histograms in Figure 6.3 shows the previous and the current approach respectively. For the same entities we also manually calculated Age-Based currency shown by the third histogram. We observe that our current approach provides higher values than the previous one. Our approach differently from the previous approach, relies on the estimated last-modification date (retrieved from the Wikpedida versions) which provides last-modification timestamps that are more reliable with respect to the timestamps provided by the triples inside the entity document. In the latter case the data producer is the only responsible for updating these triples when they modify the RDF documents.

Data freshness and semantic accuracy trade-off

We conduct the following experiments in order to evaluate if there is a correlation between data freshness and accuracy or data freshness and completeness of facts with the goal to provide a quality measure about the two data freshness dimensions.

To realize the experiments we define ten samples composed each one by thirty soccer players chosen randomly where each sample contains a set of entities that have a Age-Based currency evaluated on DBpedia defined in a specific interval (e.g., $[0, 0.1]$, $[0.1, 0.2]$, ... $[0.9, 1]$).

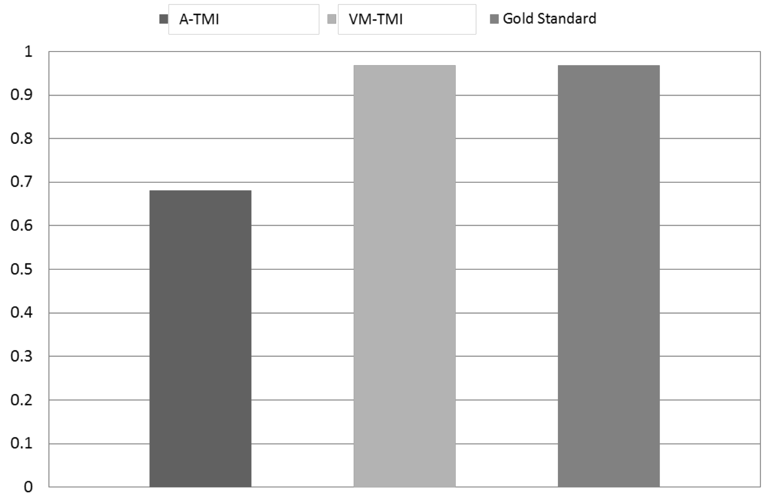


Figure 6.3: Comparison of the proposed approaches for evaluating Age-Based currency

Linear Correlation. For each of the selected entities, we measure the Age-Based currency of Wikipedia page, Age-Based and System currency of corresponding DBpedia entity, the accuracy, the completeness, and harmonic mean H between completeness and accuracy evaluated on the DBpedia entity. Then, we compute the Pearson's correlation coefficient between the measures in order to figure out if there exists a linear correlation between the quality measures.

Table 6.4: Pearson's correlation between data freshness metrics, accuracy and completeness for DBpedia

	$\beta_{Wikipedia}$	$\beta_{DBpedia}$	σ
$A(d^e)$	-0.3208	0.1596	0.4405
$C(d^e)$	-0.3066	0.2815	0.5402
$H(A(d^e), C(d^e))$	-0.3286	0.2355	0.5150

As provided in table 6.4, the results of the computations show that the Pearson's coefficients are lower than 0.75 which means that there is no linear relationship between data freshness and accuracy or data freshness and completeness. Even though the correlation between system data freshness and the other estimators is lower than 0.75, we notice that the correlation between system data freshness and completeness is higher than the other correlation.

Non-linear Correlation. In addition, we calculated the Spearman's rank correlation coefficient which is a metric that can be used to verify if there exists a non-linear correlation between the quality metrics. Table 6.5 shows the Spearman's coefficient computation between the two classes of quality measures performed on the collected soccer player entities.

TABLE 6.5: Spearman's correlation between data freshness metrics, accuracy and completeness for DBpedia

	$\beta_{Wikipedia}$	$\beta_{DBpedia}$	σ
$A(d^{e_i})$	-0.4874	-0.0713	0.5620
$C(d^{e_i})$	-0.4642	0.2349	0.8634
$H(A(d^{e_i}), C(d^{e_i}))$	-0.5351	0.0553	0.7568

The evaluation results allow us to assert that there exists a non-linear correlation between System currency and completeness because Spearman's coefficient is higher than 0.75. The correlation between the harmonic mean and the System currency is low (even if shows a value closed to the threshold), we can deduce that there exist a correlation between the two components which is driven by completeness. Finally, the complete absence of the correlation among semantic accuracy indicators and the Age-Based currency of the Wikipedia and DBpedia ones, confirms that the accuracy and completeness do not depend on the age of the document.

Regression. In order to figure out the behavior of the System currency and completeness, we compute the LOWESS (locally weighted scatterplot smoothing) function. This function is able to compute the local regression for each subset of entities and to identify, for each interval of System currency, the trend of the two distributions.

Figure 6.4 shows the LOWESS line between System currency and completeness. As provided by the graph, the local regression can not be represented by a well-known mathematical function (e.g., polynomial, exponential or logarithmic) because it has different behaviours over the three System currency intervals. Notice that for System currency values greater than 0.6, the two distributions tend to increase and for System currency values up to 0.4, the metrics tend to decrease. A particular case is shown in the interval with System currency values going from 0.4 up to 0.6 where the LOWESS is constant.

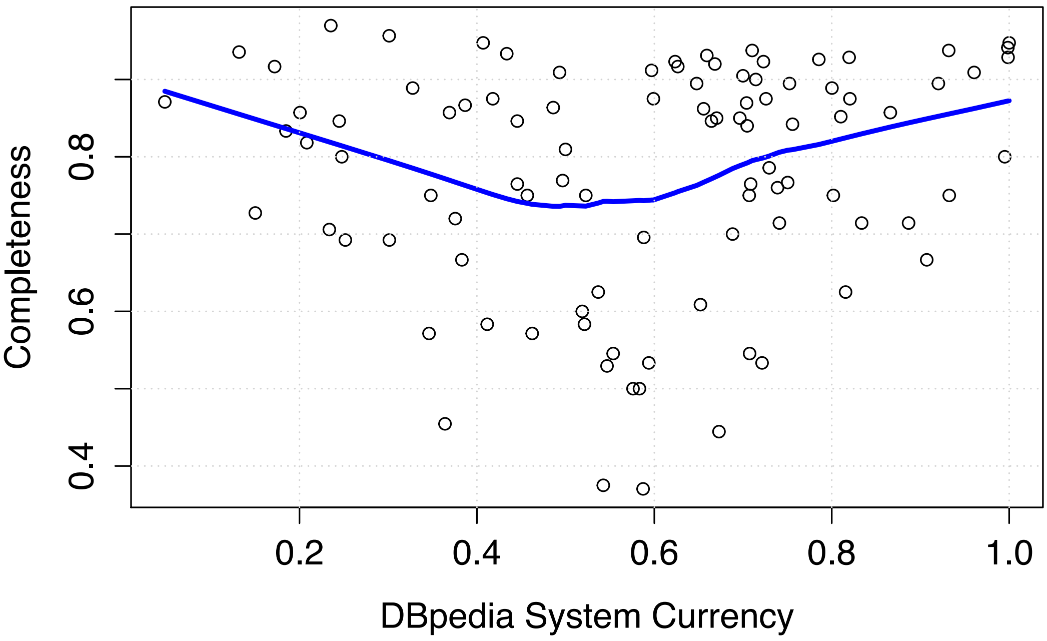


FIGURE 6.4: Local regression between System currency and completeness for entities on DBpedia

TABLE 6.6: Pearson's and Spearman's correlation between System currency intervals and completeness

σ	*Pearson's corr.*	*Spearman's corr.*
> 0.6	0.5234	0.8103
$[0.4, 0.6]$	-0.0593	-0.1330
< 0.4	-0.4103	-0.7827

In order to verify the linearity on the three entities subsets, we compute the correlation coefficients also on the identified System currency intervals. The results provided in table 6.6 show that even for the subsets there does not exist a linear correlation. In particular, there is no correlation in the central interval where the Pearson's coefficient is close to 0 and the distribution is disperse as shown in figure 6.4. Furthermore, the entities in the central interval shows that there does not exist a non-linear correlation.

Instead, the distributions increases according to a non-linear function because Spearman's coefficient is greater than 0.75 for the entities having high System currency and the lower interval distribution decrease according to a non-linear function represented by the Spearman's correlation which is lower than -0.75.

Experimental Discussion. The experimentation allows us to provide several general observations about the relation between System currency and completeness. The first one is that entities with high System currency tend to be more complete. Commonly, there is an update to the soccer player infobox each week. In particular, the two facts changing frequently in the infobox are appearances and goals scored associated with the current club. While these two facts change often, the fact that the soccer players moves to a new club can change at most two times a year. Thus, we can deduce that in a soccer player domain only two facts can change in a short time. High System currency implies low elapsed time between the last modification time of a document extracted from DBpedia and the correspondent document extracted from Wikipedia. In case the elapsed time is low, the probability that the player changed club is low too which means that only a few infobox modifications occurred. According to the formula 6.9, the completeness increases if the number of facts changing in the Wikipedia infobox is small.

The second observation concerns to the behaviour of entities with low System currency. A deep analysis of these particular entities shows that the associated Wikipedia pages have two common characteristics: (i) low frequency of modifications (approximately between six and twelve months), and (ii) refer to ex-soccer players or athletes with low popularity. The low frequency of changes in a document, typically known as *volatility* (see Chapter 4), implies a low System currency. Hence, the time elapsed between the last modification of the document extracted by DBpedia and the last modification of the associated document on Wikipedia is high. Furthermore, ex-soccer players infoboxes do not change often because they are considered to have finished their career. As a consequence of rare modifications (few changes) in the infobox attributes, such entity documents expose an high completeness.

In case of entities for which the players have low popularity, the scenario is quite different. The infobox modifications for these entities implies high completeness. Information about the unpopular players can change weekly, as well as for famous players. Therefore, if a page is not frequently updated could provide incorrect information. Consequently, we can infer that DBpedia entities of unpopular soccer players with low System currency and high completeness refers to Wikipedia pages infrequently updated, therefore, could not represent current information of real world that is also reflected in DBpedia.

At the end of such deep analysis of our experiments, we can assert that: (i) our approach for measuring data freshness based on the estimated timestamp is effective; (ii) there exist a non-linear correlation between System currency and completeness of entities; (iii) more the System currency of an entity is higher, more the associated DBpedia document has an high completeness; and (iv) entities with low System currency, that are instances of DBpedia classes of which their information can change frequently (e.g., unpopular soccer player), are associated with Wikipedia pages that could not provide real world information.

6.7 Summary

Due to their nature, Linked Open Data are not static but change frequently along time; as a consequence, in order to improve their consumption and to reduce errors related to outdated values, there is the need to estimate and understand the data freshness (age) of such data. In this chapter we proposed a framework for assessing the data freshness of LD on heterogeneous data sets. Based on an analysis of temporal meta-information used in LD as found in Billion Triple Challenge 2011, we created OntoCurrency, an ontology that integrates the most frequent temporal annotation properties used to describe temporal meta-information in the LOD cloud. The ontology allows applications, such as our framework, to collect temporal meta-information in LD data sets that use different vocabularies. Although we found that the availability of temporal entities is still limited in many domains, we obtained promising results showing the viability of the approach. Also, as the attention towards issues like provenance, change management, and data evolution is growing, we think that the availability of temporal entities to support our approach will increase.

Several are the future research directions. We plan to extend the evaluation of our approach using more queries over more data sets. An interesting research topic concerns methods to extract more accurate temporal meta-information by harvesting other sources available on the Web. However, we believe that the most interesting research direction concerns the investigation of models to predict the semantic accuracy of triples using data freshness information. While some data do not change frequently (e.g. biographies of ancient Roman imperators), other data change do (e.g. stock exchange data); data freshness has a different meaning depending on the domain and on the type of data. In this direction we are currently investigating models to measure the volatility of properties together with data freshness, following the approach proposed in [97] for relational data.

Chapter 7

Temporal Information Extraction

Metadata in general suffer from quality issues such as incompleteness or inaccuracy (Section 2.3.3) and as showed in the particular case of temporal meta-information in Chapter 5 their availability is still scarce and limited to documents rather than facts. In this chapter, we introduce an approach that combines the evidence from Web documents and Web of Data to detects the temporal validity interval, as a particular case of temporal meta-information (Section 5.2), of facts. The temporal validity interval of a fact is represented by time points indicating the starting and ending point of the interval also known as the temporal scope of the fact.

The ideas presented in this chapter are the result of joint work with Universität Leipzig, Institut für Informatik, AKSW, Germany. The discussions that lead to the development of the experimentation took place between September and December 2013. The results of our joint work have been published as shown in Appendix A.

The rest of this chapter is structured as follows: Section 7.1 describes our general approach and the system infrastructure. In Section 7.2, we describe how temporal information is extracted from the Web of documents [93] and the Web of Data. In Section 7.3, the temporal information collected is used to derive possible temporal scopes and select the most appropriate ones. We then evaluate the approach by using temporal scopes from Yago2 as gold standard and facts from DBpedia and Freebase as input in Section 7.4. Finally, we conclude and give pointers to future work (Section 7.5).

7.1 Overview

Over the last few years, the LOD Cloud has developed into a large amalgamation of diverse data sets from several domains [7]. Some of these data sets provide encyclopedic knowledge on the real world. For example, DBpedia [94] contains RDF extracted from the infoboxes of Wikipedia.[1] While some of the triples contained in the LOD Cloud are universally valid (e.g., the fact that the birth place of the soccer player Alexandre Pato is Parana', Brazil), a large portion of the facts which are referred to by the triples in the LOD Cloud are only valid within a certain time interval also known as the *time scope*. A *temporal scope* of a fact is the specification of the time during which the fact occurred. For example, DBpedia states that Alexandre Pato plays for the teams Internacional and Milan. While the semantics of the predicate `team`[2] remains a matter of discussion, manifold applications such as question answering [135], temporal reasoning and temporal information retrieval [71] require having the temporal scope of facts such as "Alexandre Pato plays for the team Milan from 2007 to 2013".

Despite several representation approaches of temporal information have been suggested, only a small amount of data sets annotate triples with their temporal validity intervals (see Chapter 5). This is partly due to the sophisticated metamodelling strategies needed to represent temporal information in RDF. As a consequence, several knowledge bases contain *volatile facts* without explicitly annotating the triples with information about their temporal scope. In practice, a temporal scope of a fact f is defined as a set of - possibly many - temporal facts of f with *disconnected* time intervals (see Section 2.2 for disconnected time intervals). For example, consider two temporal facts (⟨ `Kaka', playsFor, A.C._Milan`⟩, [2003:2009]) and (⟨ `Kaka', team, A.C._Milan`⟩, [2013: NOW])[3], the temporal scope of the fact ⟨ `Kaka', team, A.C._Milan`⟩ is [[2003:2009],[2013:NOW]].

Problem Statement. The problem addressed in this chapter can be defined as follows: for each volatile fact $f \in F$ extracted from a data set Δ, we map f to a set $TS = \{[t_{i_1}, t_{j_1}], ..., [t_{i_n}, t_{j_n}]\}$ where TS defines the temporal scopes of f and each element represents a time interval when the fact is true.

[1] http://wikipedia.org

[2] `dbo` stands for `http://dbpedia.org/ontology/`.

[3] The value NOW represents the current date.

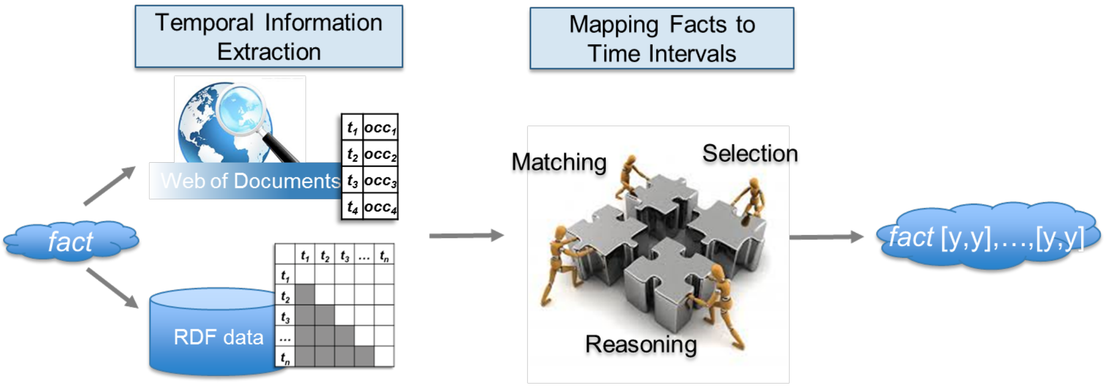


FIGURE 7.1: Approach Overview

Figure 7.1 gives an overview of our solution. Two sources can be envisaged for gathering evidence: the document Web and the Web of Data. Our approach is able to take advantage of the document Web by extending upon a fact validation approach known as DeFacto framework [93]. DeFacto allows detecting Web documents which validate a triple. In contrast to typical search engines, the framework does not just search for textual occurrences of parts of the triple, but tries to find webpages which contain the actual triple phrased in natural language. The second source of information for time scopes that our approach take advantage, is the Web of Data itself with RDF data sets that contain the facts (e.g., DBpedia or Freebase) for possible time scopes. An algorithm for combining the results extracted from Web documents with those fetched from RDF sources is devised as follows. First, the evidence extracted from Web documents is *matched* against a set of relevant time intervals to obtain a significance score for each interval. Second, a small set of more significant intervals is *selected*. Finally, the selected intervals are *merged*, when possible, by considering their mutual temporal relations. The set of disconnected intervals [3] returned by the algorithm defines the temporal scope of the fact. We also propose two *normalization* strategies that can be applied to the data extracted from Web documents before running the algorithm, to account for the significance of dates appearing in the documents corroborating the input fact.

The main contributions of this chapter are:

- We present an approach for modelling a space of relevant time intervals for a fact starting from dates extracted from RDF triples.
- We devise a three-phase algorithm for temporal scoping, i.e. for mapping facts to sets of time intervals, which integrates the previous steps via matching, selection and merging.

- Finally, we evaluate the integrated algorithm on facts extracted from DBpedia and Freebase against the Yago2 knowledge base.

7.2 Temporal Information Extraction Module

This section describes in detail the evidence extracted from Web documents (Section 7.2.1) and Web of Data (Section 7.2.2).

7.2.1 Extraction from Web Documents

DeFacto. DeFacto takes an RDF triple as input and returns a confidence value for this triple as well as possible evidence. The evidence regards Web pages and textual excerpts from those pages and meta-information on the pages which contain the input triple expressed in a natural language or textual occurrences of part of the triple.

The first task of DeFacto is to retrieve Web pages which are relevant for the given task. The retrieval is carried out by issuing several queries to a search engine. These queries are computed by verbalizing the RDF triple using natural-language patterns extracted by the Bootstrapping linked data framework (BOA) [48]. As a next step, the highest ranked Web pages for each query are retrieved, which are candidates for being sources for the input fact. Both the search engine queries as well as the retrieval of Web pages are executed in parallel to keep the response time for users within a reasonable limit. Once a Web page has been retrieved, plain text is extracted by removing HTML markup and apply our fact confirmation approach on this text. In essence, the algorithm decides whether the web page contains natural language formulations of the input fact. If no webpage confirms a fact according to DeFacto, then the system falls back on light-weight NLP techniques and computes whether the webpage does at least provide useful evidence. In addition to fact confirmation, the system computes different indicators for the trustworthiness of a webpage as presented in [109]. These indicators are of central importance because a single trustworthy webpage confirming a fact may be a more useful source than several webpages with low trustworthiness. In addition to finding and displaying useful sources, DeFacto also outputs a general confidence value for the input fact. This confidence value ranges between [0, 1] and serves as

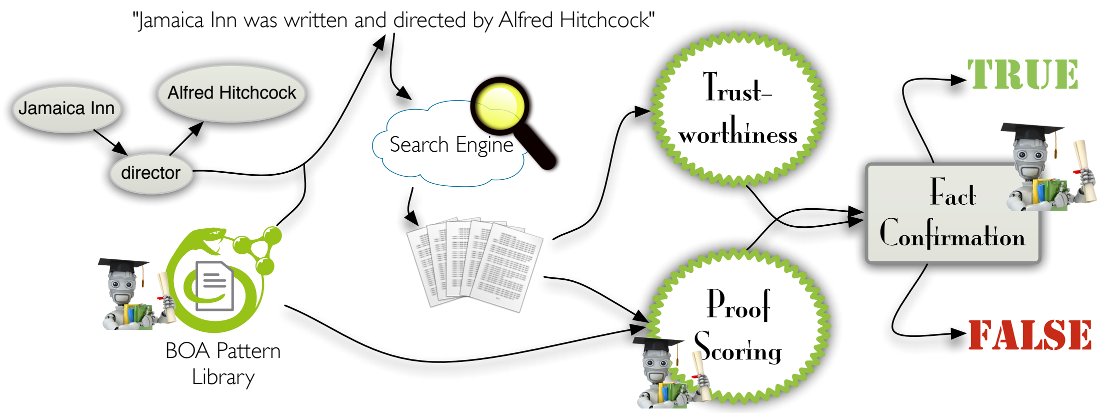


Figure 7.2: Overview of the DeFacto Framework [93].

an indicator for the user: Higher values indicate that the found sources appear to confirm the fact and can be trusted. Low values mean that not much evidence for the fact could be found on the Web and that the websites that do confirm the fact (if such exist) only display low trustworthiness. The generated provenance output can also be saved directly as RDF and abides by the PROV Ontology[4]. The source code of the DeFacto algorithms and DeFacto's user interface are open-source[5].

Temporal DeFacto. On all retrieved webpages we adopt an extended version of DeFacto, the Temporal DeFacto (for short TempDeFacto) which applies the Stanford Named Entity Tagger[6] and extracts all time points of type *Date*. The framework then examines all occurrences of the subject and object label of the input fact (or their surface forms, e.g. "Manchester United F.C." might also be called "ManU") in a proximity of less than 20 tokens. Then the tool builds a local context window of n characters before and after occ_{so} and then analyses all contained *Date* entities. Finally, TempDefacto returns a distribution of all dates and their number of occurrences in a given context. Hence, the output of TempDeFacto for a fact f can be regarded as a vector DFV over all possible time points t_i whose i^{th} entry is the number of co-occurrences of s or o with t_i. We will use the function $dfv_i(f, t_i)$ to denote the value of DFV_i for the fact f.

[4] http://www.w3.org/2011/prov/
[5] https://github.com/AKSW/DeFacto
[6] http://www-nlp.stanford.edu/software/CRF-NER.shtml

7.2.2 Information Extraction from Web of Data

To also incorporate temporal evidence from the Web of Data, we adopt an information extraction approach as follows. Given a set F of facts to map to time intervals, we first identify the set of entities E that occur as subjects for the set of facts in F. Given the entity e subject of the fact, we use the HTTP content negotiation mechanism to retrieve the entity document d^e. As an example, given the fact ⟨`Alexandre_Pato, playsFor, Milan`⟩, we extract the RDF document describing `Alexandre_Pato`. Once an entity document has been retrieved, we extract time points from the date literal triples that are contained in the entity document. As explained in Section 2.2, a *date literal triple* is a triple of the form $\langle s, p, t \rangle$, where the object t is a time point. DBpedia does not provide temporal triples (e.g., (⟨`Alexandre_Pato, playsFor, Milan`⟩, [2007:2009])), but only date literal triples (e.g., ⟨`Alexandre_Pato, years, 2006`⟩). Some of these dates refer to other facts of the same entity; however, the link between the facts containing the dates and the facts these dates were related to has been lost, neither a human nor a machine is able to understand or interpret the temporal scope of these facts. As an example, consider an excerpt of the entity document of the resource `Alexandre_Pato` in Listing 7.1. We use date literal triples available in the knowledge base under the assumption that this information can be *relevant* to define the scope of facts.

Listing 7.1: Example of an excerpt of the entity document of the resource `Alexandre_Pato`.

```
Alexandre_Pato  playsFor Sport_Club_Internacional
Alexandre_Pato  playsFor A.C._Milan.
Alexandre_Pato  birthdate 1989-09-02
Alexandre_Pato  years 2006^^xsd:integer
Alexandre_Pato  years 2007^^xsd:integer
Alexandre_Pato  years 2013^^xsd:integer
Alexandre_Pato  youthyears 2000^^xsd:integer
Alexandre_Pato  nationalyears 2007^^xsd:integer
Alexandre_Pato  nationalyears 2008^^xsd:integer
```

Given an entity e subject of a fact, we identify date literal triples in the entity document d^e and extract dates by using regular expressions, which identify standard date formats and variations. In this step we adopt the approach used in

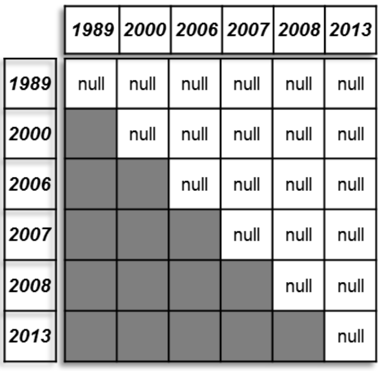
Chapter 5. We add to this set of extracted dates a date representing the *current date*. As a result of this step, each fact $f \in F$ is associated with a set of *relevant time points* T^e extracted from the RDF document describing the subject of the fact. In principle, our approach can consider dates represented at any granularity level; in the following examples and in the experiments, time is represented at the year level similarly as in other related work [98, 129].

Intuitively, we want to use the relevant time points T^e associated with an entity e to identify a set of most *relevant time intervals* for scoping facts having e as subject; in this way, we can reduce the space of all possible time intervals considered for an individual fact. The set of time intervals relevant to an entity e is defined as the set of all time intervals whose starting and ending points are members of T^e. Relevant time intervals are represented using an upper triangular matrix, i.e., a square matrix where all entries below the diagonal are 0.

Given a set T^e of relevant time points for an entity e, a relevant time interval matrix (*Relevant Interval Matrix* for short) RIM^e is an upper triangular matrix of size $|T^e| \times |T^e|$ defined as follows:

$$RIM^e = \begin{bmatrix} rim^e_{t_1,t_1} & \cdots & \cdots & rim^e_{t_1,t_n} \\ 0 & \ddots & & \\ 0 & 0 & \ddots & \\ 0 & 0 & 0 & rim^e_{t_n,t_n} \end{bmatrix} \tag{7.1}$$

Columns and rows of a relevant interval matrix RIM^e for an entity e are indexed by ordered time points in T^e; each cell $rim^e_{t_i,t_j}$ with $i, j > 0$ represents the time interval $[t_i, t_j]$, where $t_i, t_j \in T^e$. At the moment we assign a placeholder value `null` to each cell $rim^e_{i,j}$ such that $i \leq j$. In the matching phase, we will use entity-level RIMs as schemes for fact-level matrices; in these fact-level matrices `null` values will be replaced by scores that represent the significance of intervals for individual facts as shown in Figure 7.3 by the example. In this example we have considered all six time points occurring in the entity document of the resource `Alexandre_Pato`. Observe that the use of an upper triangular matrix is suitable for representing time intervals since the time intervals represented in the cells in the lower part of the matrix ($i > j$) are not valid by definition. Also note that, the cells in the diagonal of the RIM^e matrix represent time intervals whose start and end points coincide.



	1989	2000	2006	2007	2008	2013
1989	null	null	null	null	null	null
2000	■	null	null	null	null	null
2006	■	■	null	null	null	null
2007	■	■	■	null	null	null
2008	■	■	■	■	null	null
2013	■	■	■	■	■	null

FIGURE 7.3: Example of the RIM matrix for the resource `Alexandre_Pato`.

7.3 Intervals Mapping Module

The process used to provide a final mapping between a volatile fact and a set of time intervals defining its temporal scope consists of three phases: 1) Temporal Distribution-to-Time Intervals Matching, 2) Time Intervals Selection 3) Time Interval Merging.

7.3.1 Matching, Selection and Reasoning

Temporal Distribution to Time Intervals Matching. The inputs of the matching phase for a fact f that has an entity e as subject are the following: a relevant interval matrix RIM^e extracted the entity document d^e and a time distribution vector $DFV^{e,f}$ Probabilistic time distribution vectors obtained by normalization (see Section 7.3.2) can be also used as input instead of DFVs. The matching phase returns an *interval-to-fact significance matrix*, Significance Matrix (SM for short), $SM^{e,f}$ associated with the fact f. An $SM^{e,f}$ is a triangular square matrix having the same size and structure of the input RIM^e. As a next step, `null` values of a RIM^e are replaced with significance scores returned by a matching function.

In practice, to build an $SM^{e,f}$ of a fact f with subject e, we match a fact-level DFV^f associated to the fact f against an entity-level RIM^e, i.e. the matching aims to inject a time distribution vector into RIM^e by producing a significance matrix $SM^{e,f}$. The matching function $match(DFV^f, RIM^e) = SM^{e,f}$, where e is an entity and f is a fact, is given as follows:

$$sm_{i,j} = \begin{cases} 0 & \text{if } rim_{i,j} = 0 \\ \frac{\sum_{k=i}^{j} dfv(f,k)}{(j-i)+1} & \text{if } rim_{i,j} = null \land i < j \\ dfv(f,i) * w_{i,j} & \text{if } rim_{i,j} = null \land i = j \end{cases} \tag{7.2}$$

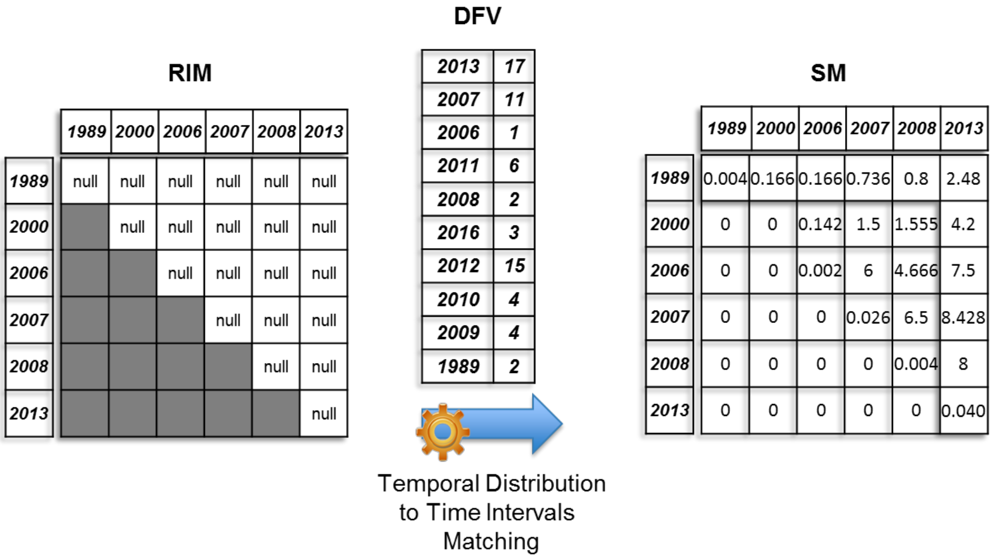
Since the denominator $(j - i) + 1$ in the formula used in case two represents the number of time points included in the interval $[i, j]$, the formula is equal to the average of DFVs for the time points contained in the interval. Since the elements in the diagonal have length equal to 1, the formula used in case three is equivalent to multiplying the score computed with the formula used in case two for a weight $w_{i,j}$; we use this weight to penalize the scores in the diagonal as we discovered that formula in case two would assign high scores to the element in the diagonal, thus favoring time intervals with length equal to 1 in the selection phase. Intuitively we want to penalize elements in the diagonal unless they are the only significant values selectable in the SM matrix. The weight is defined as inversely proportional to the difference between the length of the considered interval (equal to 1) and $length(DFV^f)$ the length of the DFV vector as follows:

$$w_{i,j} = \frac{1}{c * length(DFV^f)} \tag{7.3}$$

where c is a constant used to control the score reduction ratio applied to the elements in the diagonal of the SM matrices.

Figure 7.4 shows an example of the application of the matching phase for the fact ⟨Alexandre_Pato, playsFor, Milan⟩. The relevant interval matrix RIM containing all the possible time point extracted from the entity document of the resource Alexandre_Pato, and the time distribution vector DFV returned by Temporal DeFacto, are considered as input. The result is a SM associated with the fact ⟨Alexandre_Pato,playsFor,Milan⟩. The score for a cell $\text{sm}_{2000:2006}$ is defined as the average value of DFV for the time points between 2000 and 2006 (including the starting and ending points).

Mapping Selection. Once we have a set of significance matrices SM^{e,f_1}, ..., SM^{e,f_n}, each one associated with a fact f_i referred to e, we then select the time intervals that might be mapped to the considered facts. We propose two basic selection functions that use SMs; both functions can select more than one interval to associate with a fact f. The *top-k* function selects the k intervals that have



RIM

	1989	2000	2006	2007	2008	2013
1989	null	null	null	null	null	null
2000	▨	null	null	null	null	null
2006	▨	▨	null	null	null	null
2007	▨	▨	▨	null	null	null
2008	▨	▨	▨	▨	null	null
2013	▨	▨	▨	▨	▨	null

DFV

2013	17
2007	11
2006	1
2011	6
2008	2
2016	3
2012	15
2010	4
2009	4
1989	2

SM

	1989	2000	2006	2007	2008	2013
1989	0.004	0.166	0.166	0.736	0.8	2.48
2000	0	0	0.142	1.5	1.555	4.2
2006	0	0	0.002	6	4.666	7.5
2007	0	0	0	0.026	6.5	8.428
2008	0	0	0	0	0.004	8
2013	0	0	0	0	0	0.040

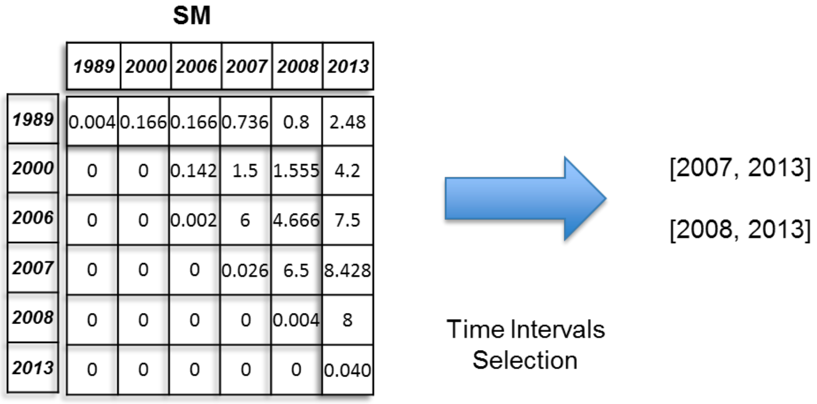
FIGURE 7.4: Example of the matching phase for the fact ⟨`Alexandre_Pato, playsFor, Milan`⟩.

best scores in the *SM* matrix. For example, consider the function *top-k* applied to the *SM* matrix of Figure 7.4 with $k = 3$. The three retrieved intervals are: [2006:2013], [2007:2013] and [2008:2013]. The *neighbor-x* selects a set of intervals whose significance score is close to the maximum significance score in the *SM* matrix, up to a certain threshold. In other terms, we define the *neighborhood of the time interval with maximum significance score* as the set of intervals whose significance scores fall in the range defined by the maximum score as upper bound and by a threshold based on a parameter x as lower bound. The threshold is linearly proportional to the maximum significance score, so that the threshold is higher when the maximum significance is higher (e.g., 0.9) and lower when the maximum significance is lower. The parametric function *neighbor-x* with an *SM* and a parameter x given as input is defined as follows:

$$neighbor(SM, x) = \left\{ [i, j] \mid sm_{i,j} \geq maxScore - \frac{x * maxScore}{100} \right\} \tag{7.4}$$

Figure 7.5 shows an example of the application of the parametric function, *neighbor-x* with x = 10. In this case, the function selects a set of intervals whose significance score is close to the maximum score `8.428`, i.e. up to the threshold score `7.58`.

The two basic functions *top-k* and *neighbor-x* can be combined into a function *neighbor-k-x* that selects the *top-k* intervals in the neighborhood of the interval with higher significance score. Observe that *neighbor-0* is equal to *top-1* for every value of the parameter x. The *neighbor-k* function behaves as a filter on the results



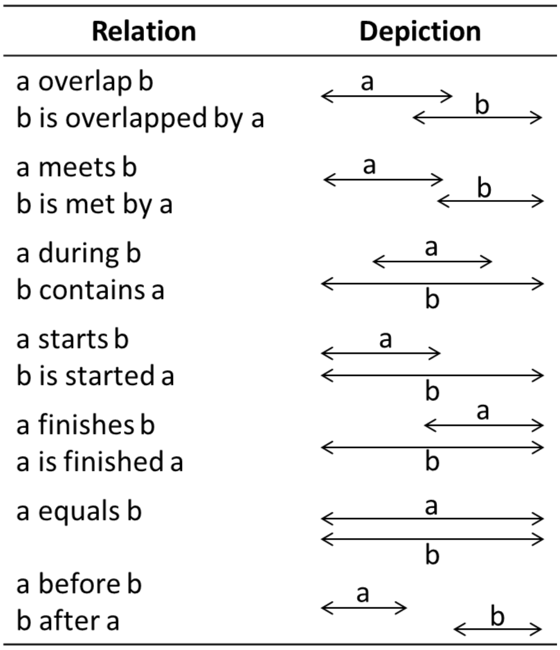
SM

	1989	2000	2006	2007	2008	2013
1989	0.004	0.166	0.166	0.736	0.8	2.48
2000	0	0	0.142	1.5	1.555	4.2
2006	0	0	0.002	6	4.666	7.5
2007	0	0	0	0.026	6.5	8.428
2008	0	0	0	0	0.004	8
2013	0	0	0	0	0	0.040

FIGURE 7.5: Example of the parametric function *neighbor-10* applied in the selection phase.

of the *top-k* function, by selecting only intervals whose significance is close enough to the most significant interval.

Interval Merging via Reasoning. Finally, we use rules based on Allen's interval algebra to merge the selected time intervals and map each fact to a set of disconnected intervals. Let a and b be two time intervals associated with a fact f and defined respectively by $[t_i, t_j]$ and $[t_h, t_k]$; we merge a and b into an interval defined by $[min(t_i, t_h), max(t_j, t_k)]$ whenever one of the following conditions shown in Figure 7.6, each one based on Allen's algebra relations [3], is verified:



Relation	Depiction
a overlap b b is overlapped by a	
a meets b b is met by a	
a during b b contains a	
a starts b b is started a	
a finishes b a is finished a	
a equals b	
a before b b after a	

FIGURE 7.6: Allen's Algebra of Time.

The temporal scope of a fact is defined by the set of disconnected time intervals mapped to it after the interval merging phase.

Figure 7.7 shows a merging example of the two intervals returned from the previous phases. In this case the relation between the two intervals is `finishes` or the inverse relation `is finished`.

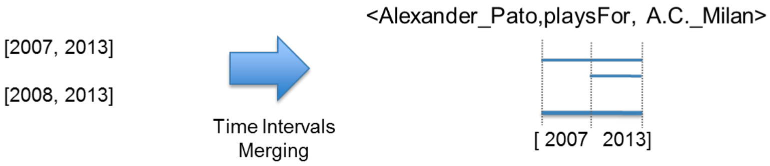


FIGURE 7.7: Example of the merging phase via reasoning.

7.3.2 Temporal Distribution Normalization

Two types of normalization functions can be envisaged: *local normalization* and *global normalization*. These functions aim to transform the output vector of temporal DeFacto (the DFV vector) into a probabilistic time distribution (PTD) vector. Here, the main idea of the local normalization is that the PTD contains the probability that the fact $\langle s, p, o' \rangle$ should be mapped to a given time point t_i. The main drawback of such a normalization is that it does not take the PTD vector for other facts $\langle s, p, o' \rangle$ into consideration. We thus defined global normalization functions that allow transforming the output of temporal DeFacto for all triples with subject s and predicate p. When normalization strategies are adopted, the PTDs are used instead of DFVs in Equation 7.2.

Local Normalization. Several approaches can be used to generate a PTD. The approach we follow is based on the frequency-based interpretation of the output of Temporal DeFacto: The i^{th} entry in DFV basically states the number of times f co-occurred with the time point t_i in a relevant document. Thus, the probability that f co-occurs with the time point t_i is:

$$PTD_i = \frac{DFV_i}{\sum\limits_{j=1}^{|T^e|} DFV_j}. \tag{7.5}$$

Global Normalization. Our approach to the computation of a global normalization was based on χ^2 statistics. Given a resource s, a predicate p and a point t_i in time, the aim of the normalization was to compute the significance of the value of DFV_i. Let E_i be the expected value of DFV_i for the time t_i, computed as average value of all DFV_i entries for the resource s over all objects of p. The significance of the time t_i for the triple $\langle s, p, o_j \rangle$ with vector DFV is then

$$\frac{(DFV_i - E_i)^2}{E_i}. \tag{7.6}$$

7.4 Experimental Evaluation

This section describes in Section 7.4.1 the experimental setup and in Section 7.4.2.

7.4.1 Experimental Setup

Methodology and Gold Standard. To evaluate the approach we acquire the temporal scopes of a population of volatile facts from three different domains and compare the results of the method against a gold standard. We use manually curated data from Yago2[7] as gold standard. We omitted all facts with `null` values, i.e. missing starting or end time. We choose Yago2 because it is one of the few large open-source knowledge bases that provides temporal annotations for a significant number of facts (714,925 time points associated with facts).

Significant parts of DBpedia[8], Freebase[9] and Yago2 are extracted from the same source which makes it possible to automatically map some facts in DBpedia or Freebase to facts in Yago2. We therefore use facts in DBpedia, and Freebase in experiments and we extract RDF data from these sources. We additionally consider the case where RIMs (see Section 7.2) are created with the time points returned by Temporal DeFacto, to simulate the case when temporal information from RDF data is not available.

Properties of Interests. The facts considered in the experiments are defined using the top three properties having the largest number of occurrences in Yago2. Table 7.1 shows the properties and the number of facts for each property.

Because we have a limit of queries sent through temporal DeFacto, which is imposed by traffic limitations of its underlying search engine, we perform the experiment on a subset of all available facts by applying some selection rules: the top

[7] `http://www.mpi-inf.mpg.de/yago-naga/yago/`

[8] `http://dbpedia.org/`

[9] `http://freebase.com/`

Table 7.1: Properties of interest and the number of facts for each property

Property	Number of facts
<ismarriedTo>	3,501
<holdsPoliticalPosition>	5,610
<playsFor>	114,367

1000 facts on the most important soccer players who are born after 1983 ($\leq$30 years old), the top 1000 facts on politicians born after 1940, and the top 500 facts on celebrities born after 1930.

Measures. In order to evaluate the accuracy of the method, we measured the degree to which the temporal scope we retrieved is correct w.r.t. the gold standard. Therefore, for each fact, we consider the degree of overlap between the retrieved intervals and the interval in the gold standard. This degree of overlap can be computed by adapting the well-known metrics of precision, recall and F_1-measure to this problem leveraging the discrete time model. Intuitively, the precision of a temporal scope can be measured by the number of time points in the temporal scope generated by our solution that fall into the time interval in the gold standard. The recall of the solution can be measured by the number of time points in the gold standard that are covered by the temporal scope.

Let $R(f)$ be the set of time points in the temporal scopes retrieved for a fact f and $\mathrm{Ref}(f)$ be the set of time points included in the reference temporal scopes for f; the following formulas capture the intuitions described above:

$$\mathit{precision}(f) = \frac{|R(f) \cap \mathit{Ref}(f)|}{|R(f)|}, \quad \mathit{recall}(f) = \frac{|R(f) \cap \mathit{Ref}(f)|}{|\mathit{Ref}(f)|}. \tag{7.7}$$

Precision and recall for a fact f can be combined as usual in F_1-measure defined as the harmonic mean of precision and recall. Note that: when $\mathit{precision}(f) = 1$, each interval in the retrieved temporal scope is included in the interval of the gold standard; when $\mathit{recall}(f) = 1$, all the time points in the interval of the gold standard are covered by the retrieved temporal scopes; when $F_1(f) = 1$ the temporal scope contains exactly the same time points as the gold standard.

Baseline. Given that no prior algorithm aims to tackle exactly the task at hand, we computed the precision, recall and F-measure that a random approach would achieve. To this end, we assumed that given the restrictions we set on the intervals within which the solutions must lie (e.g., 1983-2014 for soccer players), a random solution would simply guess for each date whether it should be part of the final solution. This serves as a lower bound for the score a temporal scoping algorithm should achieve.

7.4.2 Results and Discussion

In order to evaluate the overall accuracy of scoping facts with temporal intervals we need to set up different configurations for each component of each phase. Hence, we approximate the best configurations for some key components of the proposed approach by using genetic programming[10] based on opt4j[11], an open-source framework comprising a set of optimization algorithms. Genetic programming allows to determine an appropriate configuration of the approach. In the configuration setup we consider the interval selection functions and the merging process of the selected intervals through reasoning (see Section 7.3.1) as well as the normalizations strategies applied to the Temporal DeFacto Vectors to obtain Probabilistic Temporal Distributions (PTDs) (see Section 7.3.2).

In the first experiment, we compare the best configurations for properties of interests, i.e., (1) `isMarriedTo`, (2) `holdsPoliticalPosition` and (3) `playsFor`. The space of relevant time intervals (RIM) is built from time points collected from three different sources, i.e., Temporal DeFacto, Freebase and DBpedia.

Table 7.2: Results of best configurations for all property of interests.

Prop.	Baseline		Temp DeFacto			Freebase			DBpedia		
	#facts	F_1	Config	#facts	F_1	Config	#facts	F_1	Config	#facts	F_1
1	500	0.163	top-3	311	**0.511**	top-1 loc	213	0.477	top-1 loc	264	0.505
2	1000	0.263	top-3	709	0.586	neigh-10-2	242	0.549	neigh-10	702	**0.699**
3	1000	0.207	top-3	709	0.545	neigh-10	524	0.547	neigh-10	705	**0.600**

Table 7.2 reports for each property and for each source the best F-measure achieved by our approach. In one case, the RIM and the scores are defined with evidence retrieved only from the web of documents (TempDeFacto for short). In other two cases, the RIM is build with dates extracted from the web of data (Freebase or DBpedia) and the scores are computed by injecting evidence from the web of documents into this matrix. We observe that our approach perform much better than

[10] http://goo.gl/2ve3xP

[11] http://opt4j.sourceforge.net/

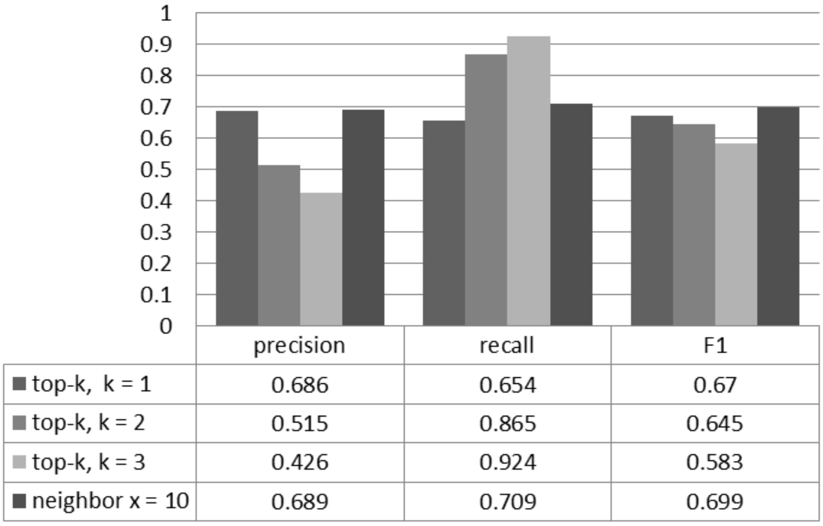
the baseline, which does not use a prior algorithm, for every property and for every source used to construct the RIMs. The best configurations is obtained for the property `holdsPoliticalPosition` with time points extracted from DBpedia and with selection function neighbor-k with $x = 10$. The configuration that extracts time points from DBpedia outperforms Freebase and Temporal DeFacto results except for the property `isMarriedTo`. The reason for this major gain can be explained with the quantity and quality of relevant time points extracted from the three sources. The problem is that Freebase and Temporal DeFacto do not provide enough time points which can prevent the effective identification of intervals. We notice that, while local normalization improves the results in one experiment (for the property `isMarriedTo`), the global normalization strategy is never optimal in any experiment. We will now compare the reasoning and selection functions.

Different Components. Table 7.3 shows the contribution of reasoning for the best configurations identified in the previous experiment. We use the full approach with and without reasoning and apply it on the three properties. We observe that enabling reasoning improves the performance of the temporal scoping of facts. This validates our motivation behind using Allen's Algebra, as it can get rid of incomplete intervals.

Table 7.3: Effect of using reasoning during temporal scoping from the three best configurations.

			With reasoning		Without reasoning	
Property	Source	Config	# facts	F_1	# facts	F_1
1	Temp DeFacto	top-3	311	0.511	505	0.467
2	DBpedia	neigh-10	702	0.699	822	0.667
3	DBpedia	neigh-10	705	0.60	977	0.563

Based on these results, we can evaluate the effect of selection functions and their application in DBpedia for the property `holdsPoliticalPosition`. Figure 7.8 compares four configurations. We observe that recall is improved when k is increased but on the other side precision decreases as the approach returns larger intervals including the correct interval and additional incorrect time points. The best precision-recall is given with the combined selection function, neighbor-k with $x = 10$.



	precision	recall	F1
■ top-k, k = 1	0.686	0.654	0.67
■ top-k, k = 2	0.515	0.865	0.645
■ top-k, k = 3	0.426	0.924	0.583
■ neighbor x = 10	0.689	0.709	0.699

FIGURE 7.8: Effect of using selection function.

7.5 Summary

In this chapter, we presented an approach for mapping volatile facts to time intervals. Our approach is hybrid and combines information from the document Web and temporal statements included in knowledge bases. We evaluated our approach on volatile facts extracted from DBpedia and Freebase by using cleaned-up temporal scopes extracted from Yago2. The cleaning was made necessary by approximately 50% of the information in that knowledge base being either incomplete or inconsistent (begin after end). This underlines the *difficulty of the task at hand*. Our approach achieved promising results, delivering approximately 70% F-measure on the facts at hand.

Chapter 8

Conclusion

The thesis analysed the concept of data quality assessment and provided a comprehensive list of quality dimensions and metrics available that can be used to assess data quality in LD. As LD is considered a dynamic environment we focused on TRQD also referred to as data freshness which is usually captured by data currency and timeliness dimensions. This thesis has investigated at a large-scale the characteristic of temporal meta-information in terms of availability, representation and diversity which are the basis for providing assessment of TRQD. We provided domain-independent techniques for the assessment of data freshness by overcoming the problems of temporal meta-information and also the peculiarities of LD.

We conclude this thesis by summing up the contributions, the significance of the achieved results and conclusions drawn thereof. Finally, we provide an outlook on further research directions for future work.

8.1 Summary of Contributions

This thesis proposes innovative techniques for the assessment of TRQD in the context of LD.

Linked Data quality dimensions. We addressed the issue of providing a comprehensive list of quality dimensions and corresponding metrics that can be used to assess data quality in LD. The dimensions and metrics were analysed by considering the challenges that LD poses with respect to previous works, in particular information systems and databases. The whole LD quality research field is evolving and cannot be considered mature enough. This can be seen also by the number of works spanned in 10 years which is rather low. One of the reasons for this could be the infancy of the research area. This study provides a broader context of quality dimensions and metrics that will be beneficial to a wide range of applications and data consumers in order to assess the quality of data sets. Among the various quality dimensions analysed we focused on TRQD since they are important for users and applications that consume dynamic data and therefore, may provide support on the validity of information.

Analysis of approaches for temporal meta-information representation. The result of this contribution is the identification of different approaches used to represent temporal meta-information in the LOD cloud. Further, due to a large-scale qualitative and quantitative analysis using the Billion Triple Challenge 2011 data set, it was possible to conclude that the availability of versioning metadata associated with documents and temporal validity of facts is still very limited. Moreover, the experiments showed that the number of temporal meta-information when available is associated with documents rather than facts. This can be due to the fact that while the representation of temporal meta-information associated with facts is somewhat more complex and can be expected to be considered in specific contexts, the temporal meta-information about the creation and modification date can be published with quite simple mechanisms. The latter would have great value, e.g., when data coming from different sources need to be integrated and fused. The investigation of temporal information and temporal meta-information is important for users or applications to be aware of the existing representation approaches of temporal information and their diffusion.

Conceptual model of versioning metadata. Based on an analysis of temporal information used in Billion Triple Challenge 2011, we developed OntoCurrency, an ontology that integrates the most frequent temporal annotation properties used to link time points or time intervals with documents or facts. The ontology allows applications, such as the assessment of TRQD, to be interoperable since it collects temporal annotation properties in LD data sets coming from different vocabularies.

Assessment techniques for Time-Related Quality Dimensions. We proposed a domain and task independent approach for assessing the data freshness of documents and triples on heterogeneous data sets. Assessing the freshness of facts when temporal meta-information associating with facts are missing, is not a straightforward task. Our framework provides the freshness assessment of RDF graphs or individual triples based on the freshness of the documents that describe the entities occurring in the graphs (triples). The assessment of documents need available temporal meta-information and when it is not possible, we provided an approach, which considered the original source adopted with a versioning mechanism where each change is identified as a revision with a time point. Although we found that the availability of temporal meta-information associated with documents and facts is still limited in many domains, we obtained promising results showing the viability of the approach. Assessment of data freshness enables consumers of the data to understand the freshness of documents or facts during the data consumption task and to help reducing the errors due to outdated data.

Improvement of temporal validity. In order to overcome the problem of incomplete or missing temporal meta-information associated with facts, we provided an approach for mapping volatile facts to their temporal validity intervals. Our approach is hybrid and combines the evidence from the Web of documents and the Web of Data (temporal triples included in knowledge bases). We evaluated our approach on volatile facts extracted from DBpedia and Freebase by using cleaned-up temporal scopes extracted from Yago2. The cleaning was made necessary by approximately 50% of the information in that knowledge base being either incomplete or inconsistent (begin after end). This underlines the *difficulty of the task at hand.* Our approach achieved promising results where in the best case achieved a maximum of 70% F-measure on the facts at hand.

8.2 Future Directions

The work presented in this thesis opens up several directions for future research.

Linked Data Quality Dimensions. As we have seen, numerous quality dimensions and metrics have been suggested in literature. But only few of them have been implemented and experimentally verified and those implemented and verified are purpose specific. To assess the quality of a data set, there should be an application which can manage the needs of users by coming up with metrics appropriate for the domain and task at hand, and decide on the relative importance of the assessment task. Providing an application for creating and understanding domain- and task-aware assessment metrics, integrating the rich work in this field, remains an open issue.

According to the definition of data quality as *fitness for use*, data quality requirements will change according to the application or use case. To assess the quality of a data set, there should be an application which can manage the data quality user requirements by coming up with metrics appropriate for the domain and task at hand, and decide on the relative importance of the assessment task. The overall objective of the application is to assess data quality independently on the domain or task. Providing an application which has the ability to self-adapt according to changes occurring internally to the execution environment (data quality user requirements change according to the domain or to the specific task) is a research field in software engineering. Software applications having the ability to self-adapt at runtime to handle resource variability, changing user needs, and system intrusions or faults are known as *self-adaptive systems* [32]. Such systems must configure and reconfigure themselves, continually tune and optimize themselves in cases when user needs are changing. In particular, we are interested in one of the identified essential views of self-adaptation which is requirement engineering for self-adaptive systems. One of the main challenges that self-adaptation poses is that when designing a self-adaptive system, we cannot assume that all adaptations are known in advance - that is, we cannot anticipate requirements for the entire set of possible environmental conditions and their respective adaptation specifications [32]. The application must be able to gather and elaborate data quality user requirements depending on the task and the domain of the quality assessment process.

Time-Related Quality Dimensions. It is obvious that domain-independent assessment techniques, as discussed here, provide an intrinsic quality value about the data set taken into consideration. However, the problem of assessing TRQD of a data set cannot be considered completely solved since the problem of the temporal meta-information availability still holds over. A possible direction could be to extract more accurate temporal meta-information from the Web documents. But in the next future probably not all data sets will have a correspondent source on the Web. The assessment of TRQD provided some initial results, and we would welcome more focused research into other specific areas. For instance, consider the study of change frequency of facts. While some data do not change frequently (e.g. biographies of ancient Roman imperators), other data does (e.g. stock exchange data). Note that, not all apparent changes occurring in a document are significant ones (e.g., the change of an image, the correction of a typos). Therefore, instead of estimating the change frequency of documents, which is deeply studied in the Web documents [34], we propose a study on change frequency of facts in the Web of Data, which is an area that is still largely unexplored. We are currently working in this direction, by investigating models to predict the change frequency of facts.

Temporal Information Extraction First, it would be interesting to apply the hybrid approach of extracting temporal information within further domains. As different domains need to be evaluated, we plan to enlarge the actual gold standard. Second, we plan to compare our approach with the approaches proposed in the Text Analysis Conference(TAC)[1] in the KBP research area in particular in the Temporal Slot Filling Task [81]. As we already mentioned our approach is completely unsupervised and the NLP techniques we adopt are softer with respect to the NLP techniques employed in TAC 2010 KBP task or other similar works [39]. Moreover, our approach investigates how to complement evidence retrieved from texts with evidence from the Web of Data. Finally, we plan to develop temporal query answering applications that exploit temporal information rather than providing only TRQD assessment.

[1] http://www.nist.gov/tac/

Appendix A

Published Work

Parts of the work presented in this thesis have been published in international journals and the proceedings of international conferences and refereed workshops. Publications relating to this work are listed below:

International Journals

- Zaveri, Amrapali, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sren Auer. "Quality Assessment for Linked Data: A Survey." Submitted to Semantic Web Journal (2013).
- Carlo Batini, Marco Comerio, Anisa Rula, Gianluigi Viscusi. "From Data Quality to Big Data Quality." Submitted to Journal of Database Management (2014).

International Conferences

- Anisa Rula, Matteo Palmonari, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, Jens Lehmann and Lorenz Bhmann. "Hybrid Acquisition of Temporal Scopes for RDF Data." In Extended Semantic Web Conference (ESWC) 2014.
- Rula, Anisa, Matteo Palmonari, Andreas Harth, Steffen Stadtmller, and Andrea Maurino. "On the diversity and availability of temporal information in linked open data." In International Semantic Web Conference (ISWC) 2012, pp. 492-507. Springer Berlin Heidelberg, 2012.

Refereed Workshops

- Rula, Anisa, Matteo Palmonari, and Andrea Maurino. "Capturing the age of linked open data: Towards a dataset-independent framework." In Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on, pp. 218-225. IEEE, 2012.
- Julia Hoxha, Anisa Rula, Basil Ell. "Towards Green Linked Data". COLD 2011.
- Anisa Rula. "DC Proposal: Towards Linked Data Assessment and Linking Temporal Facts". In International Semantic Web Conference (ISWC) 2011, pp. 341-348.

Bibliography

[1] Turtle - terse RDF triple language, W3C team submission, 2008. http://www.w3.org/TeamSubmission/turtle/.

[2] Akoka, J., Berti-Equille, L., Boucelma, O., Bouzeghoub, M., Comyn-Wattiau, I., Cosquer, M., Goasdoué-Thion, V., Kedad, Z., Nugier, S., Peralta, V., et al. A framework for quality evaluation in data integration systems. In *ICEIS (3)* (2007), pp. 170–175.

[3] Allen, J. F. Maintaining knowledge about temporal intervals. *Communications of the ACM 26*, 11 (1983), 832–843.

[4] Alonso, O., Strötgen, J., Baeza-Yates, R., and Gertz, M. Temporal Information Retrieval: Challenges and Opportunities. In *1st Temporal Web Analytics Workshop at WWW* (2011), pp. 1–8.

[5] Alvestrand, H. Rfc 3066 - tags for the identification of languages. Tech. rep., IETF, 2001.

[6] Artz, D., and Gil, Y. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web 5*, 2 (2007), 58–71.

[7] Auer, S., Lehmann, J., and Ngomo, A.-C. N. Introduction to linked data and its lifecycle on the web. In *5th RR* (2011), pp. 1–75.

[8] Baader, F., Diageo, C., McGuinness, D., Nardi, D., and Patel-Schneider, P., Eds. *The Description Logic Handbook*. Cambridge, 2003.

[9] Ballou, D., Wang, R., Pazer, H., and Tayi, G. K. Modeling Information Manufacturing Systems to Determine Information Product Quality. *Management Science* (1998), 462–484.

[10] Bartlett, J., Kotrlik, I., and Higgins, C. Organizational Research: Determining Appropriate Sample Size in Survey Research. *Information Technology, Learning, and Performance Journal* (2001), 43.

[11] Batini, C., Cappiello, C., Francalanci, C., and Maurino, A. Methodologies for data quality assessment and improvement. *ACM Computing Surveys 41*, 3 (2009).

[12] Batini, C., and Scannapieco, M. *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[13] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein, L. A. OWL Web Ontology Language Reference. Tech. rep., W3C, http://www.w3.org/TR/owl-ref/, February 2004.

[14] Beckett, D. RDF/XML Syntax Specification (Revised). Tech. rep., World Wide Web Consortium, 2004. http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/.

[15] Berman, F. Got data?: a guide to data preservation in the information age. *Communications of the ACM 51*, 12 (2008), 50–56.

[16] Berners-Lee, T. RFC 2396: Uniform Resource Identifiers (URI). Tech. rep., MIT, 1998. http://www.rfc-archive.org/getrfc.php?rfc=2396.

[17] Berners-Lee, T., and Connolly, D. Notation3 (n3): A readable rdf syntax. Tech. rep., W3C, 1 2008. http://www.w3.org/TeamSubmission/n3/.

[18] Berners-Lee, T., Hendler, J., Lassila, O., et al. The semantic web. *Scientific American 284*, 5 (2001), 28–37.

[19] Bizer, C. *Quality-Driven Information Filtering in the Context of Web-Based Information Systems*. PhD thesis, Freie Universität Berlin, March 2007.

[20] Bizer, C., and Cyganiak, R. Quality-driven information filtering using the WIQA policy framework. *Web Semantics 7*, 1 (Jan 2009), 1 – 10.

[21] Bizer, C., Cyganiak, R., and Heath, T. How to publish Linked Data on the Web. linkeddata.org Tutorial, 2008.

[22] Bizer, C., Heath, T., and Berners-Lee, T. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems* (2009), 1–22.

[23] Böhm, C., Naumann, F., Abedjan, Z., Fenz, D., Grütze, T., Hefenbrock, D., Pohl, M., and Sonnabend, D. Profiling linked open data with ProLOD. In *ICDE Workshops* (2010), IEEE, pp. 175–178.

[24] Bonatti, P. A., Hogan, A., Polleres, A., and Sauro, L. Robust and scalable linked data reasoning incorporating provenance and trust annotations. *Journal of Web Semantics 9*, 2 (2011), 165 – 201.

[25] Bouzeghoub, M., and Peralta, V. A framework for analysis of data freshness. In *IQIS* (2004), pp. 59–67.

[26] Bovee, M., Srivastava, R., and Mak, B. A Conceptual Framework and Belief-Function Approach to Assessing Overall Information Quality. In *6th International Conference on Information Quality* (September 2001).

[27] Brickley, D., and Guha, R. V. RDF vocabulary description language 1.0: RDF schema. Tech. rep., W3C, 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[28] Bright, L., and Raschid, L. Using latency-recency profiles for data delivery on the web. In *Proceedings of the 28th International Conference on Very Large Data Bases* (2002), VLDB '02, VLDB Endowment, pp. 550–561.

[29] Brown, A. R. Wikipedia as a data source for political scientists: Accuracy and completeness of coverage. *PS Political Science and Politics 44* (3 2011), 339–343.

[30] Burton-Jones, A., Storey, V. C., Sugumaran, V., and Ahluwalia, P. A semiotic metrics suite for assessing the quality of ontologies. *Data Knowl. Eng. 55*, 1 (Oct. 2005), 84–102. http://dx.doi.org/10.1016/j.datak.2004.11.010.

[31] Carroll, J. Signing RDF graphs. Tech. rep., HPL-2003-142, HP Labs, 2003.

[32] Cheng, B. H., De Lemos, R., Giese, H., Inverardi, P., Magee, J., Andersson, J., Becker, B., Bencomo, N., Brun, Y., Cukic, B., et al. Software engineering for self-adaptive systems: A research roadmap. In *Software engineering for self-adaptive systems*. Springer, 2009, pp. 1–26.

[33] Cho, J., and Garcia-Molina, H. The Evolution of the Web and Implications for an Incremental Crawler. In *The 26th VLDB* (2000), pp. 200–209.

[34] Cho, J., and Garcia-Molina, H. Estimating frequency of change. *ACM Trans. Internet Technol. 3*, 3 (Aug. 2003), 256–290.

[35] Correndo, G., Salvadores, M., Millard, I., and Shadbolt, N. Linked timelines: Temporal representation and management in linked data. In *1st International Workshop on Consuming Linked Data at International Semantic Web Conference*. 2010.

[36] Crocker, D. H. Standard for the Format of ARPA Internet Text Messages. RFC 822, 1982.

[37] de Sompel, H. V., Sanderson, R., Nelson, M. L., Balakireva, L., Shankar, H., and Ainsworth, S. An HTTP-Based Versioning Mechanism for Linked Data. In *3rd Linked Data on the Web Workshop at WWW* (2010).

[38] Demter, J., Auer, S., Martin, M., and Lehmann, J. LODStats – an extensible framework for high-performance dataset analytics. In *EKAW* (2012), LNCS, Springer.

[39] Derczynski, L., and Gaizauskas, R. Information retrieval for temporal bounding. In *4th ICTIR* (2013), ACM, pp. 29:129–29:130.

[40] Ell, B., Vrandečic, D., and Simperl, E. Labels in the web of data. In *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I* (Berlin, Heidelberg, 2011), ISWC'11, Springer-Verlag, pp. 162–176.

[41] Fallside, D. C., and Walmsley, P. XML Schema Part 0: Primer Second Edition. World Wide Web Consortium, 2004.

[42] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. Hypertext transfer protocol – http/1.1

(rfc 2616). Request For Comments, 1999. available at http://www.ietf.org/rfc/rfc2616.txt, accessed 7 July 2006.

[43] Flemming, A. Quality characteristics of linked data publishing datasources, 2010. http://sourceforge.net/apps/mediawiki/trdf/index.php?title=Quality_Criteria_for_Linked_Data_sources.

[44] Fürber, C., and Hepp, M. SWIQA - a semantic web information quality assessment framework. In *ECIS* (2011).

[45] Gamble, M., and Goble, C. Quality, trust, and utility of scientific data on the web: Towards a joint model. In *ACM WebSci* (June 2011), pp. 1–8.

[46] Gangadharan, G. R., Weiss, M., D'Andrea, V., and Iannella, R. Service license composition and compatibility analysis. In *ICSOC* (2007), pp. 257–269.

[47] Gangemi, A., Catenacci, C., Ciaramita, M., and Lehmann, J. Modelling ontology evaluation and validation. In *The Semantic Web: Research and Applications*, Y. Sure and J. Domingue, Eds., vol. 4011 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 140–154. http://dx.doi.org/10.1007/11762256_13.

[48] Gerber, D., and Ngomo, A.-C. N. Extracting Multilingual Natural-Language Patterns for RDF Predicates. In *18th EKAW* (2012), Springer.

[49] Gertz, M. Data quality on the web. http://www.dagstuhl.de/files/Reports/03/03362.pdf Retrieved 15/02/2014, 2003.

[50] Gil, Y., and Artz, D. Towards content trust of web resources. *Web Semantics 5*, 4 (December 2007), 227 – 239.

[51] Gil, Y., and Ratnakar, V. Trusting information sources one citizen at a time. In *ISWC* (2002), Springer-Verlag, pp. 162 – 176.

[52] Glimm, B., and Ogbuji, C. Sparql 1.1 entailment regimes. Tech. rep., 2012. http://www.w3.org/TR/sparql11-entailment/.

[53] Golbeck, J. Using trust and provenance for content filtering on the semantic web. In *Workshop on Models of Trust on the Web at the 15th World Wide Web Conference* (2006).

[54] Golbeck, J., Parsia, B., and Hendler, J. Trust networks on the semantic web. In *Cooperative Intelligent Agents* (2003).

[55] Grant, J., and Becket, D. Rdf test cases - n-triples. Tech. rep., W3C Recommendation, 2004. http://www.w3.org/TR/rdf-testcases/#ntriples.

[56] Groth, P. T., Gil, Y., Cheney, J., and Miles, S. Requirements for provenance on the web. *IJDC 7*, 1 (2012), 39–56.

[57] Guéret, C., Groth, P., Stadler, C., and Lehmann, J. Assessing linked data mappings using network measures. In *ESWC* (2012).

[58] Gutiérrez, C., Hurtado, C. A., and Vaisman, A. A. Temporal RDF. In *The 2nd ESWC* (2005), pp. 93–107.

[59] Gutiérrez, C., Hurtado, C. A., and Vaisman, A. A. Temporal rdf. In *2nd Extended Semantic Web Conference* (2005), pp. 93–107.

[60] Halpin, H., Hayes, P., McCusker, J. P., McGuinness, D., and Thompson, H. S. When owl:sameas isn't the same: An analysis of identity in linked data. In *Proceedings of the 9th International Semantic Web Conference (ISWC)* (2010), vol. 1, pp. 53–59.

[61] Hansen, M. T., and Von Oetinger, B. Introducing t-shaped managers. knowledge management's next generation. *Harvard business review 79*, 3 (2001), 106–16.

[62] Hartig, O. Trustworthiness of data on the web. In *STI Berlin and CSW PhD Workshop, Berlin, Germany* (2008).

[63] Hartig, O. Provenance Information in the Web of Data. In *2nd Linked Data on the Web Workshop at WWW* (2009).

[64] Hartig, O., and Freytag, J. C. Foundations of traversal based query execution over linked data (extended version). *CoRR abs/1108.6328* (2011). http://dblp.uni-trier.de/db/journals/corr/corr1108.html#abs-1108-6328.

[65] Hartig, O., and Zhao, J. Using Web Data Provenance for Quality Assessment. In *Workshop on Semantic Web and Provenance Management at ISWC* (2009).

[66] Hayes, P. RDF Semantics. Recommendation, World Wide Web Consortium, 2004. http://www.w3.org/TR/2004/REC-rdf-mt-20040210.

[67] Heath, T., and Bizer, C. *Linked Data: Evolving the Web into a Global Data Space*, 1st ed. No. 1:1 in Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan and Claypool, 2011, pp. 1 – 136.

[68] Heinrich, B., and Klier, M. Assessing data currency - a probabilistic approach. *J. Information Science 37*, 1 (2011), 86–100.

[69] Hitzler, P., Krtzsch, M., Parsia, B., Patel-Schneider, P. F., and Rudolph, S. OWL 2 Web Ontology Language Primer. W3C Recommendation, World Wide Web Consortium, October 2009.

[70] Hobbs, J., and Pan, F. An Ontology of Time for the Semantic Web. *Processings of the ACM Transactions on Asian Language Information* (2004), 66–85.

[71] Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence 194* (2013), 28–61.

[72] Hogan, A., Harth, A., Passant, A., Decker, S., and Polleres, A. Weaving the pedantic web. In *LDOW* (2010).

[73] Hogan, A., Harth, A., Passant, A., Decker, S., and Polleres, A. Weaving the Pedantic Web. In *3rd Linked Data on the Web Workshop at WWW* (2010).

[74] Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A., and Decker, S. Searching and browsing linked data with swse: the semantic web search engine. *Web semantics: science, services and agents on the world wide web 9*, 4 (2011), 365–401.

[75] Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., and Decker, S. An empirical survey of Linked Data conformance. *Journal of Web Semantics* (2012).

[76] Horton, M. R. Standard for Interchange of USENET Messages. RFC 850, Internet Engineering Task Force, 1983.

[77] Hovy, D., Fan, J., Gliozzo, A., Patwardhan, S., and Welty, C. When did that happen?: linking events and relations to timestamps. In *13th EACL* (2012).

[78] ISO 8601. Data Elements and Interchange Formats-Information Interchange-Representation of Dates and Times, 2004.

[79] Jacobi, I., Kagal, L., and Khandelwal, A. Rule-based trust assessment on the semantic web. In *International conference on Rule-based reasoning, programming, and applications series* (2011), pp. 227 – 241.

[80] Jacobs, I., and Walsh, N. *Architecture of the World Wide Web, Volume One*. The World Wide Web Consortium (W3C), Dec. 2004. http://www.w3.org/TR/webarch/.

[81] Ji, H., Grishman, R., and Dang, H. Overview of the TAC2011 knowledge base population track. In *TAC 2011 Proceedings Papers* (2011).

[82] Juran, J. *The Quality Control Handbook*. McGraw-Hill, New York, 1974.

[83] Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., and Hogan, A. Observing linked data dynamics. In *ESWC* (2013), pp. 213–227.

[84] Käfer, T., Umbrich, J., Hogan, A., and Polleres, A. Towards a Dynamic Linked Data Observatory. In *5th Linked Data on the Web Workshop at WWW* (2012).

[85] Klyne, G., and Carroll, J. J. Resource description framework (RDF): Concepts and abstract syntax. World Wide Web Consortium, Recommendation REC-rdf-concepts-20040210, February 2004.

[86] Knight, S., and Burn, J. Developing a framework for assessing information quality on the World Wide Web. *Information Science 8* (2005), 159 – 172.

[87] Knight, S., and Burn, J. Developing a framework for assessing information quality on the world wide web. *Informing Science 8* (2005).

[88] Koubarakis, M., and Kyzirakos, K. Modeling and Querying Metadata in the Semantic Sensor Web: the Model stRDF and the Query Language stSPARQL. *The 7th ESWC* (2010), 425–439.

[89] Krötzsch, M., and Speiser, S. Sharealike your data: Self-referential usage policies for the semantic web. In *International Semantic Web Conference (1)* (2011), pp. 354–369.

[90] Lassila, O., and Swick, R. R. Resource Description Framework (RDF) Model and Syntax Specification. W3c recommendation, W3C, February 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

[91] Lee, H. T., Leonard, D., Wang, X., and Loguinov, D. IRLbot: Scaling to 6 Billion Pages and Beyond. In *The 17th WWW* (2008), pp. 427–436.

[92] Lee, Y. W., Strong, D. M., Kahn, B. K., and Wang, R. Y. AIMQ: a methodology for information quality assessment. *Information Management 40*, 2 (2002), 133 – 146.

[93] Lehmann, J., Gerber, D., Morsey, M., and Ngonga Ngomo, A.-C. DeFacto - deep fact validation. In *11th ISWC* (2012), Springer, pp. 312–327.

[94] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal* (2013). Under review.

[95] Lei, Y., Uren, V., and Motta, E. A framework for evaluating semantic metadata. In *4th International Conference on Knowledge Capture* (2007), no. 8 in K-CAP '07, ACM, pp. 135 – 142.

[96] Leo Pipino, Ricahrd Wang, D. K., and Rybold, W. *Developing Measurement Scales for Data-Quality Dimensions*, vol. 1. M.E. Sharpe, New York, 2005.

[97] Li, P., Dong, X. L., Maurino, A., and Srivastava, D. Linking temporal records. *Proceedings of the VLDB Endowment 4*, 11 (2011), 956–967.

[98] Ling, X., and Weld, D. S. Temporal information extraction. In *25th AAAI* (2010).

[99] Mannino, M. V., and Walter, Z. A framework for data warehouse refresh policies. *Decision Support Systems 42*, 1 (2006), 121–143.

[100] Margaritopoulos, T., Margaritopoulos, M., Mavridis, I., and Manitsaris, A. A conceptual framework for metadata quality assessment. *Universitätsverlag Göttingen 104* (2008).

[101] Mecella, M., Scannapieco, M., Virgillito, A., Baldoni, R., Catarci, T., and Batini, C. Managing data quality in cooperative information systems. In *Confederated International Conferences DOA, CoopIS and ODBASE* (2002), pp. 486–502.

[102] Meilicke, C., and Stuckenschmidt, H. Incoherence as a basis for measuring the quality of ontology mappings. In *3rd International Workshop on Ontology Matching (OM) at the ISWC* (2008).

[103] Mendes, P., Mühleisen, H., and Bizer, C. Sieve: Linked data quality assessment and fusion. In *LWDM* (March 2012).

[104] Mendes, P. N., Mühleisen, H., and Bizer, C. Sieve: Linked Data Quality Assessment and Fusion. In *2nd Int'l Workshop on Linked Web Data Mgmt at EDBT* (2012).

[105] Miller, P., Styles, R., and Heath, T. Open data commons, a license for open data. In *LDOW* (2008).

[106] Moreau, L. The foundations for provenance on the web. *Foundations and Trends in Web Science 2*, 2–3 (2010), 99–241.

[107] Mostafavi, M., G., E., and Jeansoulin, R. Ontology-based method for quality assessment of spatial data bases. In *International Symposium on Spatial Data Quality* (2004), vol. 4, pp. 49–66.

[108] Motik, B., Patel-Schneider, P. F., Parsia, B., Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., and Smith, M. OWL 2 web ontology language: Structural specification and functional-style syntax. Last call working draft, W3C, 2008. http://www.w3.org/2007/OWL/draft/owl2-syntax/.

[109] Nakamura, S., Konishi, S., Jatowt, A., Ohshima, H., Kondo, H., Tezuka, T., Oyama, S., and Tanaka, K. Trustworthiness analysis of web search results. In *11th ECDL* (2007).

[110] Naumann, F. *Quality-Driven Query Answering for Integrated Information Systems*, vol. 2261 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[111] Nunes, S., Ribeiro, C., and David, G. Using neighbors to date web documents. In *Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management* (New York, NY, USA, 2007), WIDM '07, ACM, pp. 129–136.

[112] Orlandi, F., and Passant, A. Modelling provenance of {DBpedia} resources using wikipedia contributions.

[113] Palavitsinis, N., Manouselis, N., and Sanchez-Alonso, S. Metadata quality in digital repositories: Empirical results from the cross-domain transfer of a quality assurance process. *Journal of the Association for Information Science and Technology* (2014), n/a–n/a.

[114] Panziera, L., Comerio, M., Palmonari, M., De Paoli, F., and Batini, C. Quality-Driven Extraction, Fusion and Matchmaking of Semantic Web API Descriptions. *J. Web Eng. 11*, 3 (2012), 247–268.

[115] Panziera, L., Comerio, M., Palmonari, M., Paoli, F. D., and Batini, C. Quality-driven extraction, fusion and matchmaking of semantic web api descriptions. *J. Web Eng. 11*, 3 (2012), 247–268.

[116] Peralta, V. *Data Quality Evaluation in Data Integration Systems*. PhD thesis, University of Versailles (France) & University of the Republic (Uruguay), 2006.

[117] Pernici, B., and Scannapieco, M. Data quality in web information systems. *Journal on Data Semantics I* (2003), 48–68.

[118] Pipino, L., Lee, Y. W., and Wang, R. Y. Data quality assessment. *Communications of the ACM 45*, 4 (2002).

[119] Popitsch, N., and Haslhofer, B. DSNotify - A Solution for Event Detection and Link Maintenance in Dynamic Datasets. *Web Semantics* (2011), 266–283.

[120] Prud'hommeaux, E., and Seaborne, A. SPARQL query language for RDF, W3C recommendation. Tech. rep., World Wide Web Consortium, January 2008. http://www.w3.org/TR/rdf-sparql-query/.

[121] Raggett, D., Hors, A. L., and Jacobs, I. HTML 4.01 specification. W3C Recommendation, W3C - World Wide Web Consortium, December 1999. http://www.w3.org/TR/html401/.

[122] Rodriguez, A., McGrath, R., Liu, Y., and Myers, J. Semantic Management of Streaming Data. *2nd International Workshop on Semantic Sensor Networks at ISWC* (2009).

[123] SalahEldeen, H., and Nelson, M. L. Carbon dating the web: Estimating the age of web resources. *CoRR abs/1304.5213* (2013).

[124] Scheuermann, A., Motta, E., Mulholland, P., Gangemi, A., and Presutti, V. An empirical perspective on representing time. In *Proceedings of the seventh international conference on Knowledge capture* (2013), ACM, pp. 89–96.

[125] Schober, D., Barry, S., Lewis, E. S., Kusnierczyk, W., Lomax, J., Mungall, C., Taylor, F. C., Rocca-Serra, P., and Sansone, S.-A. Survey-based naming conventions for use in OBO foundry ontology development. *BMC Bioinformatics 10*, 125 (2009).

[126] Shekarpour, S., and Katebi, S. Modeling and evaluation of trust with an extension in semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web 8*, 1 (March 2010), 26 – 36.

[127] Shin, B. An exploratory investigation of system success factors in data warehousing. *Journal of the Association for Information Systems 4* (2003).

[128] Stvilia, B., Gasser, L., Twidale, M. B., Shreeves, S. L., and Cole, T. W. Metadata quality for federated collections. In *IQ* (2004), I. N. Chengalur-Smith, L. Raschid, J. Long, and C. Seko, Eds., MIT, pp. 111–125.

[129] Talukdar, P. P., Wijaya, D. T., and Mitchell, T. Coupled temporal scoping of relational facts. In *5th WSDM* (2012), pp. 73–82.

[130] Tappolet, J., and Bernstein, A. Applied temporal rdf: Efficient temporal querying of rdf data with sparql. In *6th Extended Semantic Web Conference* (2009), pp. 308–322.

[131] Tappolet, J., and Bernstein, A. Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In *The 6th ESWC* (2009), pp. 308–322.

[132] Tayi, G. K., and Ballou, D. P. Examining data quality. *Commun. ACM 41*, 2 (Feb. 1998), 54–57. http://doi.acm.org/10.1145/269012.269021.

[133] Theodoratos, D., and Bouzeghoub, M. Data currency quality factors in data warehouse design. In *In Proc. of the Int. Workshop on Design and Management of Data Warehouses (DMDW'99* (1999).

[134] Umbrich, J., Hausenblas, M., Hogan, A., Polleres, A., and Decker, S. Towards Dataset Dynamics: Change Frequency of Linked Open Data Sources. In *3rd Linked Data on the Web Workshop at WWW* (2010).

[135] Unger, C., Bühmann, L., Lehmann, J., Ngomo, A.-C. N., Gerber, D., and Cimiano, P. Sparql template-based question answering. In *21st WWW* (2012).

[136] UzZaman, N., and Allen, J. F. Trips and trios system for tempeval-2: Extracting temporal information from text. In *SemEval* (2010), ACL, pp. 276–283.

[137] Villata, S., and Gandon, F. Licenses compatibility and composition in the web of data. In *COLD* (2012).

[138] Vrandecic, D. *Ontology Evaluation.* PhD thesis, Karlsruher Instituts für Technologie (KIT), 2010.

[139] Wand, Y., and Wang, R. Y. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM 39*, 11 (1996), 86–95.

[140] Wang, R. Y. A product perspective on total data quality management. *Communications of the ACM 41*, 2 (Feb 1998), 58 – 65.

[141] Wang, R. Y., and Strong, D. M. Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems 12*, 4 (1996), 5–33.

[142] Wang, Y., Dylla, M., Ren, Z., Spaniol, M., and Weikum, G. Pravda-live: interactive knowledge harvesting. In *21st CIKM* (2012), ACM, pp. 2674–2676.

[143] Wang, Y., Zhu, M., Qu, L., Spaniol, M., and Weikum, G. Timely YAGO: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia. In *The 13th EDBT* (2010), pp. 697–700.

[144] Welty, C., Fikes, R., and Makarios, S. A Reusable Ontology for Fluents in OWL. *Frontiers in Artificial Intelligence and Applications* (2006), 226.

[145] Yu, Y., and Heflin, J. Extending functional dependency to detect abnormal data in rdf graphs. In *The International Semantic Web Conference*, vol. 7031. Springer Berlin Heidelberg, 2011, pp. 794–809.

[146] Zaveri, A., Kontokostas, D., Sherif, M. A., Bühmann, L., Morsey, M., Auer, S., and Lehmann, J. User-driven quality evaluation of dbpedia. In *To appear in Proceedings of 9th International Conference on Semantic Systems, I-SEMANTICS '13, Graz, Austria, September 4-6, 2013* (2013), ACM. http://svn.aksw.org/papers/2013/ISemantics_DBpediaDQ/public.pdf.

[147] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., and Auer, S. Quality assessment for linked open data: A survey (under review). *Semantic Web Journal* (2012). This article is still under review.