

# Big Data Semantics

Paolo Ceravolo · Antonia Azzini · Marco Angelini · Tiziana Catarci ·  
Philippe Cudré-Mauroux · Ernesto Damiani · Alexandra Mazak ·  
Maurice Van Keulen · Mustafa Jarrar · Giuseppe Santucci · Kai-Uwe  
Sattler · Monica Scannapieco · Manuel Wimmer · Robert Wrembel ·  
Fadi Zaraket

the date of receipt and acceptance should be inserted later

**Abstract** Big Data technology has discarded traditional data modelling approaches as no longer applicable to distributed data processing. It is, however, largely recognised that Big Data impose novel challenges in data and infrastructure management. Indeed, multiple components and procedures must be coordinated to ensure a high level of data quality and accessibility for the application layers, e.g. data analytics and reporting. In this paper, the third of its kind co-authored by mem-

bers of IFIP WG 2.6 on Data Semantics, we propose a review of the literature addressing these topics and discuss relevant challenges for future research. Based on our literature review, we argue that methods, principles, and perspectives developed by the Data Semantics community can significantly contribute to address Big Data challenges.

## 1 Introduction

The term “Big Data” is widely used to designate a discontinuity in data processing and analytics [1,2]. The early literature described this discontinuity using the “5 Vs” storyline that highlights the unprecedented data Volume, Velocity (e.g. in terms of input data rate), Variety (in terms of data types), as well as non-uniform Veracity and Value of today’s applications [3–6]. In other words, data-intensive applications require a data processing rate that may exceed the resources available on a single node and this condition is in general difficult to predict when dealing with online data streams [7]. On-demand elastic computing platforms, such as Amazon AWS [8], and distributed processing frameworks, such as Apache Hadoop and Spark [9], have been developed as a technological solution for addressing these scalability issues. The attention of the research community has, accordingly, focused on processing functions [10,11] and execution performance [12], giving less attention to other key features of information management, like for example reuse, verifiability, and modularity.

Data and infrastructure management represent recurring challenges [5,13] for Big Data. Due to the heterogeneous nature of distributed data, Big Data applications do not make use of traditional data mod-

---

P. Ceravolo  
Università Degli Studi di Milano, Italy  
A. Azzini  
Consortium for the Technology Transfer, C2T, Italy  
M. Angelini  
SAPIENZA University of Rome, Italy  
T. Catarci  
SAPIENZA University of Rome, Italy  
P. Cudré-Mauroux  
University of Fribourg, Switzerland  
E. Damiani  
EBTIC, Khalifa University, Abu Dhabi, UAE  
A. Mazak  
Vienna University of Technology, Wien, Austria  
M. Van Keulen  
University of Twente, The Netherlands  
M. Jarrar  
Birzeit University, Palestine  
G. Santucci  
SAPIENZA University of Rome, Italy  
K. Sattler  
TU Ilmenau, Germany  
M. Scannapieco  
Directorate for Methodology and Statistical Design  
Italian National Institute of Statistics (Istat), Rome, Italy  
M. Wimmer  
Vienna University of Technology, Wien, Austria  
R. Wrembel  
Poznan University of Technology, Poland  
F. Zaraket  
American University of Beirut, Lebanon

elling [14]. Distributed datasets and streams may consist of unstructured, semi-structured or highly structured but still non-relational data items such as time series or nested records to which traditional data modeling techniques are problematic. Clearly, heterogeneous and/or weakly structured data make it difficult to design schemata in advance [15]. In addition, Big Data datasets may be processed only a few times or even once per use-case, making it too expensive to load them into a database management system. In turn, data heterogeneity is taken to the extreme in data-intensive applications involving Cyber Physical Systems.

Another issue in Big Data representation is finding data formats suitable for feeding a variety of algorithms ranging from simple filters and aggregates to complex machine learning models. Metadata, i.e. data that describe other data or systems, are essential for any data management activity including data integration and cleaning, maintaining consistency and freshness, and above all efficient querying. Traditional relational databases and data warehouses offer metadata repositories and metadata management as a built-in feature [16]. However, a metadata standard has not yet been developed for Big Data technologies.

Moreover, Big Data involves multi-party processes, with different legal frameworks that apply to the data provider, the data collector, and the data miner. Data management have then become a major area of interest for data protection. In fact, collecting evidence on the procedures and practices applied, using continuous monitoring and assurance components [17], is today essential.

In summary, Big Data still lacks a comprehensive, sound approach to data management. Rethinking information management in the context of Big Data technologies is a primary topic for future research. A challenge that involves the whole process of the Big Data pipeline, i.e. the set of tasks required to drive Big Data computations. Documentation, reconfiguration, data quality assurance, and verification are examples of crucial tasks not easily supported in the current landscape of Big Data technologies.

The aim of this paper is to explain how *Data Semantics* can support Big Data management. Based on this perspective, we propose a review of the literature addressing this issue and discuss directions for future research. In particular, our attention will focus the FAIR principles [18] recommending procedures that generate Findable, Accessible, Interoperable, and Re-usable data or metadata.

Based on our literature review, as well as on the collective vision of the members of the IFIP WG 2.6 on Data Semantics we will discuss how methods, princi-

ples, and perspectives developed by the Data Semantics community can contribute to address Big Data challenges. In this respect, this paper ideally continues the tradition of WG 2.6 collective contributions [19,20].

The structure of this paper is the following. In Section 2 and Section 3 we introduce our discussion by distinguishing different levels of representation that can be adopted in Big Data management, taking into account the different stages composing a Big Data pipeline. In Section 4 we develop the central discussion of this paper. Then, in Section 5, we review the research perspectives that emerge from our discussion and, finally, in Section 6, we draw the conclusions.

## 2 Data Semantics Dimensions

Achieving the full potential of Big Data analytics requires realizing a reconciliation between data distribution and data modeling principles [14,21]. An improper data representation may reduce the accuracy of analytics or even invalidate their results [22]. It can also impact the cost of execution of analytics. Besides, a mismatch on the abstraction level adopted by different data sources may occur even when they rely on a shared data dictionary [23].

*Data Semantics* refers to the “meaning and meaningful use of data” [24], i.e. the effective use of a data object for representing a concept or object in the real world. Such a general notion interconnects a large variety of applications.

A historic achievement of the database community was **Representing Data** via suitable schemata. Unfortunately, Big Data deal with evolving heterogeneous data that make it difficult, or even impossible, to identify a data schema prior to data processing. Solutions for integrating and querying schema-less data have then received much attention [25].

However, since schema-based representation techniques cannot be directly applied to describe Big Data, more and more attention is being directed to **Representing Metadata** within data-intensive applications. Managing a large volume of heterogeneous and distributed data requires definition and continuous updating of metadata describing different aspects of semantics and data quality, such as data documentation, provenance, trust, accuracy, and other properties [26]. Recently, the IEEE has launched an initiative aimed at fostering standardisation in Big Data management<sup>1</sup>.

---

<sup>1</sup> The IEEE Big Data Governance and Metadata Management (BDGMM) group ([http://standards.ieee.org/develop/indconn/BDGMM\\_index.html](http://standards.ieee.org/develop/indconn/BDGMM_index.html)) aims at enabling data integration among heterogeneous datasets from diversified domain repositories and make data

A further application of Data Semantics principles to Big Data involves **Modeling Data Processes** and flows, i.e. representing the entire pipeline making data representation shareable and verifiable. This may furthermore include the relationships interconnecting the different stages of a pipeline, for example, processing data stream requires to select appropriate data preparation modules and analytics.

Finally, we also underline the relevance of **Data Quality** aspects. Each phase of the Big Data pipeline has its own quality tasks that must be taken into account in order to get high-quality outputs from Big Data analytics.

### 3 The Big Data Pipeline

It is well recognized that Big Data impacts the entire workflow guiding the execution of analytics. The complexity of Big Data technologies and the variety of competences required to design applications relying on them [27] have emphasized the notion of Big Data pipelines, as well as the relevance of systems for managing [28] and documenting them. A pipeline is the coordination of different tasks, integrating different technologies, to achieve a specific solution [29,30]. The Hadoop Stack includes, for example, services related to five areas: *Data Source*, *Data Format*, *Data Stores*, *Data Staging*, and *Data Processing* [5]. Among the most comprehensive overviews on reference components of a pipeline, the authors of [31] propose the following steps: *Data Extraction*, *Data Loading and Pre-processing*, *Data Processing*, *Data Analysis*, *Data Loading and Transformation*, as well as *Data Interfacing and Visualization*. Ardagna et al. [32] focus on languages for the explicit representation of a pipeline and propose the following areas: *Data Representation*, *Data Preparation*, *Data Processing*, *Data Analytics*, and *Data Visualization*.

Without claiming to be comprehensive, in the followings, we present a pipeline inspired by [33]. The terminology introduced here will guide our discussion in Sections 4 and 5.

- **Data Acquisition and Recording.** Big Data arise from one or several data generating sources that must be interpreted in the appropriate way, filtering out irrelevant data before starting any processing.
- **Data Extraction and Annotation.** Frequently, the data collected will not be in a format suitable

for analysis. A data extraction process is then required to format data in a structured form suitable for analysis, for example by extracting structured data from semi-structured or unstructured contents.

- **Data Preparation and Cleaning.** Records may be inaccurate or corrupted. Detecting, correcting, and removing such records from a dataset is a crucial step. A preparation stage may also be required to increase the obfuscation level of the data, for preserving privacy, intellectual property, or strategic knowledge.
- **Data Integration and Aggregation.** Given the heterogeneity of distributed data, it is not enough to merely load them into a distributed storage. A certain degree of integration, summarization, and standardization is necessary.
- **Data Processing and Querying.** Methods for querying and mining Big Data are fundamentally different from the traditional statistical analysis. This is because the impact of data distribution and performance requirements on algorithms, hence on processing behavior, is significant.
- **Data Interpretation and Reporting.** The complexity arising from Big Data technologies renders a simple monitoring and reporting insufficient as a means for interpretation and evaluation of results. A rigorous interpretation requires multiple stages to verify the assumptions that allow drawing safe conclusions.

Figure 1 offers a synthetic view on the pipeline adopted as a reference for this paper.

---

discoverable, accessible, and usable through an actionable standard infrastructure.

Reference	Stages of the BD Pipeline						Data Semantics Dim.				
	Data Acquisition and Recording	Data Extraction and Annotation	Data Preparation and Cleaning	Data Integration and Aggregation	Data Processing and Querying	Data Analysis and Modelling	Data Interpretation and Reporting	Data	Metadata	Data Quality	Process Representation
Markl [14]						x		x			
Ardagna et al. [34]			x	x	x	x	x				x
Azzini et al. [23]		x						x			
Smith et al. [26]				x					x		
Liao et al. [35]	x								x		
Duggan et al. [36]				x				x			
Sowmya et al. [37]						x		x			
Zhou et al. [38]	x								x		
Akoush et al. [39]	x								x		
Glavic [40]	x								x		
Berti-Equille et al. [41]	x									x	
Kläs et al. [42]		x							x		
Daiber et al. [43]		x							x		
Shin et al. [44]		x	x						x		
Chiticariu et al. [45]		x							x		
Fuhring et al. [46]		x								x	
Bondiombouy et al. [47]				x				x			
Bergamaschi et al. [48]				x					x		
Ramakrishnan et al. [49]				x					x		
Masseroli et al. [50]				x					x		
Scannapieco et al. [51]				x						x	
Gualtieri et al. [52]				x				x			
Liu et al. [53]			x						x		
Gulzar et al. [54]			x						x		
De Wit [55]			x							x	
Zardetto et al. [56]			x							x	
Gonzalez et al. [57]					x			x			
Junghanns et al. [58]					x			x			
Yu et al. [59]					x			x			
You et al. [60]					x			x			
Hagedorn et al. [61]					x			x			
Kornacker et al. [62]					x			x			
Costea et al. [63]					x			x			
Schätzle et al. [64]					x			x			
Cudré-Mauroux et al. [65]					x			x			
Appice et al. [66]						x		x			
Khare2015 et al. [67]						x		x			
Poggi et al. [68]						x		x			
Um et al. [69]						x		x			
Poole et al. [16]						x			x		
Smith et al. [26]						x			x		
Giese et al. [70]				x		x			x		
UNECE [71]						x				x	
Severin et al. [72]							x		x		
Mezghani et al. [73]							x		x		
Ginsberg et al. [74]							x			x	
Sculley et al. [75]					x	x					x
Chang et al. [76]					x			x			
Plale et al. [77]			x					x			
Terrizzano et al. [78]			x					x			
Teradata [79]			x					x			
Scannapieco et al. [80]	x	x								x	
Agrawal et al. [81]						x				x	
Liu et al. [82]							x				

Reference	Stages of the BD Pipeline							Data Semantics Dim.			
	Data Acquisition and Recording	Data Extraction and Annotation	Data Preparation and Cleaning	Data Integration and Aggregation	Data Processing and Querying	Data Analysis and Modelling	Data Interpretation and Reporting	Data	Metadata	Data Quality	Process Representation
Damiani et al. [83]			x					x			
Doan et al. [84]				x					x		
Flood et al. [85]				x					x		
Haryadi et al. [86]		x								x	
Benedetti et al. [87]	x	x							x		
Ford et al. [88]		x							x		
Haas et al. [89]			x						x		
Cabot et al. [90]				x					x		
Voigt et al. [91]				x					x		
Soylu et al. [92]				x					x		
McKenzie et al. [93]				x					x		
Habib et al. [94]			x						x		
Magnani et al. [95]			x	x					x		
Van Keulen [96]						x				x	
Andrews et al. [97]						x					x
Sparks et al. [98]						x					x
Meng et al. [99]						x					x
Flunkert et al. [100]						x					x
Baylor et al. [101]						x					x

**Table 1** References organized by Stages of the Big Data Pipeline and Data Semantics Dimensions

## 4 The Contribution of Data Semantics to Big Data Management

In the next sections we organize our discussion following the stages of the Big Data pipeline described in Section 3. A summary of this discussion is contained in Table 1 that lists the main references included in this discussion and maps them to the most pertinent stages of the Big Data Pipeline and Data Semantics dimensions.

### 4.1 Data Acquisition and Recording

Provenance represents a major issue and has been recognised as a key requirement for Big Data applications [81]. Provenance is about tracking the transformation process that generated a certain piece of data. This often implies recording a variety of metadata, for example about the execution environment that generated a transformation, the authority that issued a data set, or a quality measure that was recorded for this

data set. Such metadata can then be exploited in support of a variety of applications, including debugging, trust, assurance, and security [40].

Malik et al. [38] present an approach for recording provenance in a distributed environment. Provenance metadata is recorded each time a file version is generated by a node that also maintains a summary of the provenance metadata it generated and a link to the nodes that shared files with him. This way, each node is generating a local view on provenance but the whole system organises a graph of provenance metadata that supports queries over the provenance across node boundaries. Following a similar approach, provenance systems that capture provenance data generated by MapReduce computations within the Hadoop framework were developed [35, 39].

The “regulatory barrier”, i.e. concerns about violations of data access, sharing, and custody regulations when using Big Data, and the high cost of obtaining legal clearance for their specific scenario has been rec-

ognized as a major factor hindering Big Data technologies [102].

According to the EU directive on data protection [103], personal data is not simply data stored in the entries of a repository but any information that can be inferred by processing these entries in combination to others. The information that can be inferred by a service provider that manages user profiles, which involve for instance personal interests, personal knowledge, skills, goals, behavior, social, or environmental context the user interacts with [83]. Stakeholders are more and more aware of these capabilities and can perceive risks in releasing data to third parties. Thus, it is essential to design technologies capable of natively reporting how Big Data are stored and processed in the different stages of their life-cycle.

For the acquisition/recording phase, it is important to observe that the Big Data user is often different from the Big Data provider. This poses the very relevant challenge of the authoritativeness of the Big Data source. All the dimensions of the Trust cluster [80] should be taken into account, namely: believability, reliability, and reputation. In [41], the need for a framework for truth discovery on the Web is claimed and an indication of possible solutions to such a challenge are also provided. In Figure 1, the **source quality evaluation** task takes this aspect into account.

## 4.2 Data Extraction and Annotation

As noted in [86], the adoption of Big Data platforms requires an overhaul of traditional metadata management processes to cope with potentially higher data dimensionality, higher acquisition rates, and multipartite data. In particular, data extraction techniques have to be upgraded to face these new realities. In the current context, data extraction relates to the process of extracting (semi-) structured data from largely unstructured content such as natural language texts and images. Many recent works have been focusing on extracting entities from documents. Entities are textual elements that are of interest to a data extraction task, such as persons, places, companies, or events. Often, such entities are first identified in text and then linked to their counterpart in a knowledge base. In addition to entities, relations connecting a pair of entities and/or annotations adding labels to chunks of text are often extracted. Many systems have been developed for this purpose, such as SystemT [45], Textmarker [104], Dualist [105], MMAX [106], or BRAT [107].

Most of such systems address text in a single language only. However, support for multiple language

is essential for data extraction and annotation techniques. Contextual semantic analysis [87] is gathering interest as a promising technique to tackle multilingual computational tasks. The idea is to represent documents as contextual graphs that are sub-graphs of reference knowledge bases, such as the linked open data cloud [108] and DBPedia [109]. A contextual graph is represented in a semantic context vector which in turn can be used to compare against other documents. The paper [110], for instance, successfully leverages contextual semantic analysis to build semantic relations between nominals across multi-lingual (English, German, and Russian) and multi-genre documents. Deep canonical correlation analysis leverages multilingual semantics (English/German) to improve word embeddings [111].

Data extraction methods often use statistical measures based on parallel corpora that include training text from one language with translation into one or more other languages. Approaches to parallel corpora construction range from manually constructing parallel documents to automated efforts that search for similar documents across the web [43, 112]. Wikipedia is one of the largest semi-structured and semantically annotated parallel corpora. Interestingly, the multilingual aspect features more than information enrichment across cultures and local preferences. It also exhibits conflicts and edit wars that reflect priorities, regional, and cultural differences [113]. Current repositories offering multilingual resources include the Linguistic Data Consortium, the European Language Resource Association, the University of Oxford Text Archive, GlobalPhone, Tractor, EuroWordnet, GlobalWordNet, IXA pipes, and MUL-Text.

Statistical techniques perform poorly for under-resourced languages such as Chinese, Vietnamese, and Arabic. Consequently, domain specific techniques that use hand-crafted grammars [114], expert knowledge that identify mappings rules [115] and language-invariant features such as mappings of part-of-speech (POS) and ontology-based annotations [116] have been introduced to boost the performance of statistical automatic translation and information retrieval.

Technical reports, news articles, scientific papers, web pages, and literary books contain multilingual texts that in general respect presentation rules, layout structures, and linguistic grammar. However, several some niche languages have sporadic structure if any. For example, SMS and chat messages use proprietary abbreviations and no sentence structure. Notes in electronic medical records [88] are similar except that medical doctors compose them while paying attention to the patient and not the keyboard or the screen. This results in more typing errors. Similar to under-resourced and

morphologically rich languages, such documents require preprocessing and expert knowledge for information extraction and machine translation tasks.

For the extraction/annotation phase, it is important to extract or derive all quality metadata that could support the subsequent processing phases. Quality metadata are very much source- and task-dependent. As an example, if wishing to perform an analysis of Twitter data that needs to take into account the location information associated to a specific tweet, then such location related metadata should be part of the extraction. A specific quality characterization of such metadata should be performed, for instance to assess if location is accurate by linking it to a geographical dictionary. An interesting challenge is to develop tools that automatically or semi-automatically allow to annotate data with quality scores (see [46]). In Figure 1, the **quality metadata extraction and annotation task** includes such activities.

#### 4.3 Data Preparation and Cleaning

Data preparation for Big Data analytics encounters many obstacles. “Analysts report spending upwards of 80% of their time on problems in data cleaning” [89]. A central data quality problem is handling semantic duplicates: two or more records that actually represent the same real-world entity. Besides probabilistic evaluation techniques being inherently imperfect, also merging records inevitably leads to conflicts. As a consequence, the result of a data integration process is inherently fuzzy. In particular if dealing with unstructured data, such as data harvested from web sites (e.g., [44]) or from social media (e.g., [94]), where we are requested to deal with natural language that is inherently ambiguous, moreover, crowd-sourced data may be unreliable or incomplete.

One important development is using metadata to represent uncertainty. A nice survey on uncertainty in data integration is [95]. In essence, the idea is to model all kinds of data quality problems as uncertainty in the data [96, 117]. This way, uncertain data can be stored and managed in a probabilistic database [118], or by aggregating metadata in the form of possibilistic assertions [119].

Semantic duplicates are almost never detected with absolute certainty unless both records are identical. Therefore, there is a grey area of record pairs that may or may not be semantic duplicates. Instead of requiring a manual inspection and an absolute decision, a probabilistic database can simply directly store the indeterministic deduplication result [120]. Furthermore, the resulting data can be directly queried and analyzed.

Imperfect results of information extraction can be represented as uncertain data as well. For example, the mentioning of a named entity like “Paris” can refer to the capital of France, but also to more than 60 other places called “Paris” or even to a person. Probabilistic databases can easily represent such results directly as well as any related data, such as population.

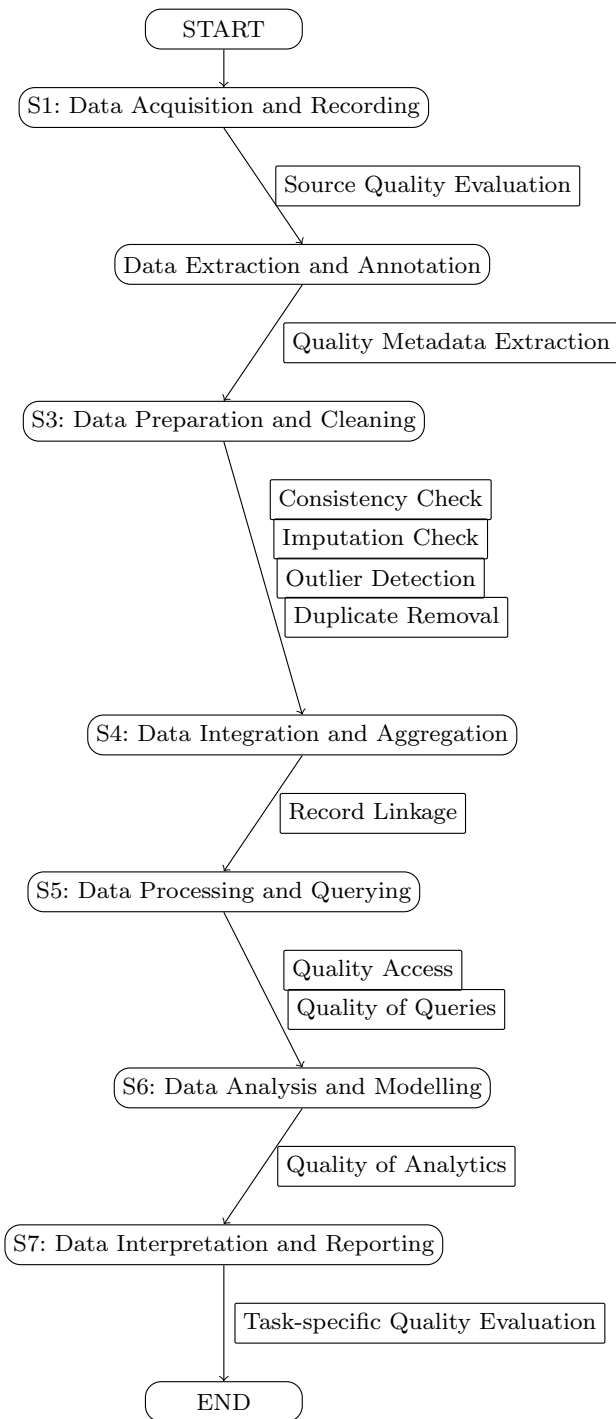
When resources do not allow for a full investigation of detailed semantics, these techniques can be used as a way to *cast doubt* over the data. *Data profiling* can quickly give both valuable insights into the semantics of data as well as into the source’s quality problems [121]. For example, various types of functional dependencies can be mined from the data itself [122]. They specify constraints, or rather expectations, that the context or application imposes on the data. Any violations of these may uncover exceptional situations (semantics) or errors (data quality).

These techniques have achieved right now a limited adoption in Big Data commercial frameworks. However, various scholars have addressed the topic. In [53] a metadata generation algorithm is used for enriching data with description about context and usage patterns. Metadata are then exploited to efficiently clean and remove inconsistencies in a dataset or in a set of linked datasets.

The data preparation and cleaning phase can be detailed in terms of sub-tasks. The most significant ones are: (a) localization and correction of inconsistencies; (b) localization and correction of incomplete data and (c) localization of outliers (see [80] for an overview of these approaches for traditional data). In terms of Big Data sources, the techniques for performing the cleaning tasks are source- and task-specific. As an example, the Automatic Identification System (AIS) is a system that permits to detect ships, by providing the location and status information of ships over a radio channel. In [55] the authors describe the usage of these data for maritime statistics and provide a good detail of the very specific cleaning tasks for these data, such as the removal of glitches due to errors in the identification system. The specific tasks considered in Figure 1 are: **consistency checks, imputation procedures, outliers detection, duplicate detection**.

#### 4.4 Data Integration and Aggregation

Data Management in Big Data cannot be simply resolved by an efficient storage and query infrastructure. Data Integration is equally important. Typically, Big Data are being integrated by means of Data Lake (DL) architectures [52, 77–79]. A DL is a repository that



**Fig. 1** Main stages of a Big Data Pipeline and related Data Quality Tasks.

stores a vast amount of heterogeneous data in their original formats, e.g., relational tables, Web tables, XML and its variants, texts, images, sounds, videos, graphs, time series. In most DL deployments, the data storage layer is based on a distributed file system - HDFS or

GFS, and data is processed in parallel (typically, by a MapReduce-like parallelisation patterns) [5,123].

In [79], the authors advocate the following four stages of implementing a DL: (1) learning, planning, and implementing mechanisms of ingesting data, (2) improving the quality of data, by applying ETL/ELT processes [123], (3) developing mechanisms for jointly querying and analyzing data stored in an enterprise Data Warehouse (DW) and the DL, (4) augmenting the DL with the so-called 'enterprise capabilities', i.e., governance, compliance, security, and auditing.

Often, DLs need to ingest data from multiple data sources spread over the Web, in the framework of applications such as sentiment analysis, trend analysis, adverse events analysis, or others. In such a context, it is important to be able to: (1) discover the most relevant data sources, (2) figure out their structures, content, and quality, and finally (3) plug the discovered data sources of interest into a DL integration architecture, to ingest their content. These processes raise challenges in developing methods for discovering and profiling [124] data sources on the Web.

The processes that ingest data into a DL do not change the structure of the data being uploaded but store it in their original formats - this feature is known as *no schema on write*. The structure of data is however important for applications that read, process, and analyze the data. Therefore, such application has to discover and understand data formats on the fly - this feature is known as *schema on read*. To this end, rich and well-organized metadata are needed to provide a precise description of the data. In order to ease the process of querying a DL, a kind of global schema on various data structures in the DL is needed. This leads us towards an observation that a revised data modeling approach is needed that would be able to capture the heterogeneity of data and that would give a foundation for querying such data.

Having built a DL and having ingested data, another issue is to keep the DL up-to-date, similarly as a traditional DW. The content of a DL has to be refreshed either periodically or in (near) real-time (for application areas like fraud detection, installation monitoring). Refreshing a DL is much more complex than a traditional DW. It is due to the fact that multiple data sources are available on the Web with limited capabilities of accessing them. Therefore, detecting their data and structural changes is challenging. Moreover, as data formats are much more than simple tables, incorporating incremental updates to data stored in a DL is more difficult. To this end, new change detection techniques and efficient refreshing algorithms need to be developed, as even in-



cremental refreshing may upload much larger data volumes than in a traditional DW architecture.

Recall that the content of a DL typically stores dumps from various data sources (including the Web) whose data quality is often poor. Therefore, advanced data cleaning, augmentation, and imputation techniques (e.g., [125–128]) need to be applied to the content of a DL.

Today, integrating data in a data lake is handled manually, for the most part, resulting in slow and cumbersome processes. In fact, data preparation and integration is often considered as the most time-consuming part of a data science project<sup>2</sup>. As such, it is urgent to develop better solutions to this problem.

To provide solutions for the aforementioned challenges, some research and development works have already been done, mainly in the field of querying a DL and providing a schema-like view on the DL.

Three architectures that allow to execute SQL queries on a Hadoop-based data lakes were identified in [52]:

- *Pure SQL for Hadoop* - such an engine includes an SQL interpreter that is capable of using Hive-like external tables and exploiting metadata about their definitions. Examples of such engines include among others: Hive, Spark, Drill, Kylin, Impala, and Presto.
- *Boosted SQL for Hadoop* - an engine supports more advanced capabilities of query parsing, optimizing, and scheduling. Examples of such engines include among others: Actian Vortex, HP Vertica, IBM Big SQL, JethroData, Pivotal HAWQ, Phoenix, Splice Machine, and Virtuoso.
- *Database+ for Hadoop* - an access to data stored in Hadoop is realized directly from a fully-functional DBMS, by means of the standardized SQL provided by this DBMS. To this end, a Hadoop data source is linked to the DBMS by means of external tables. Examples of such solutions include among others: Microsoft PolyBase, Oracle Big Data SQL, Teradata QueryGrid, EMC Greenplum, SAP Vora. Such technologies offer a means for querying jointly an enterprise data warehouse and a data lake.

An overall architecture of a data lake (discussed in this section) is shown in Figure 2.

Recently, another Big Data integration architecture was proposed in [36], which is called a *polystore*. The main idea behind it is to organize data sets into the

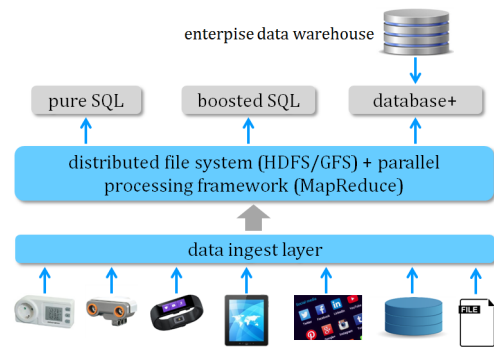


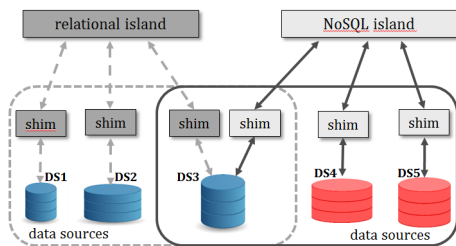
Fig. 2 An overall architecture of a data lake

so-called *islands of information*. An island of information is defined as a collection of storage engines accessed with the same query language. For example, in a data lake, several relational islands, graph islands, XML islands (each managed by a separate system) can be stored and all of them can be part of a polystore. An island exhibits to a user its data model and provides a common query language, similarly as a mediator [129]. The language and data model are mapped, by a software module called a *shim*, into a specific language and model of a data management system running the island. This functionality is similar to what provides a wrapper [129].

Query processing on an island is executed as follows. First, an island query is expressed in the island native language. Second, the query is parsed into an abstract syntax tree (AST). Third, the AST is decomposed into partial queries - one query for one DS in the island. Fourth, partial queries are sent to appropriate shims. Next, each shim translates its partial query into a query in a native language of a data source. Finally, the partial queries are executed in their proper data sources. The query language of an island (proposed in [36]) was extended with 2 clauses, namely: *scope* - for specifying in which island a query is to be executed and *cast* - for indicating an output data model and for copying data between islands. Multi-island queries are also allowed by means of shims from different islands.

An example architecture of a polystore system is shown in Figure 3. It is composed of two information islands: a relational one and a NoSQL one. The first island is composed of relational data sources  $DS_1$ ,  $DS_2$ , and  $DS_3$ , each of which has its dedicated shim that exposes a relational interface to the island. The second island is composed of NoSQL data sources  $DS_4$  and  $DS_5$ . Their shims expose a NoSQL interface to the NoSQL island. Notice that  $DS_3$  is shared by both islands through 2 shims, i.e., it can be queried from both islands.

<sup>2</sup> see for instance <https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html>



**Fig. 3** An example architecture of the polystore with two islands of information

A lighter approach to integrating widely different data in a data lake is to resort to a *knowledge graph* as the main mediation layer. The knowledge graph in question can be public (e.g., WikiData<sup>3</sup> or DBPedia<sup>4</sup>) or private (e.g., Google’s Knowledge Graph<sup>5</sup>). The idea is then to link all instances in the data lake to instances in the knowledge graph, and to retrieve all relevant pieces of data for a given application by issuing search queries (leveraging ad-hoc object retrieval [130] techniques). The overall approach is termed *entity-centric* [131] in the sense that entities in the knowledge graph play the role of the global schema in a federated database.

Metadata are crucial to integrate and combine different datasources [84]. For example, the services provided by different medical centers may be recorded differently in different data sets but they all refer to a same set of standard codes that are ferried as metadata of the dataset.

Making file metadata scalable is one of the top challenges addressed in the literature [49]. In connection to this, scalable data integration methods have been also investigated [26, 48].

There are sectors where data integration is pursued centers the emergence of a global standard for identifying data objects. For example the legal entity identifier (LEI) system is aimed at providing a unified approach for representing financial data [85]. A metadata model supporting data integration was proposed to link genomic feature data to their associated experimental, biological and clinical use [50]. However, global standards alone are insufficient for high-quality integration [132]. In general, data integration is a difficult task to be automated as it often requires knowledgeable input from domain experts. Thus, some authors suggest to explore the use of advanced interactive approaches such as active learning and crowd-sourcing [133].

The data integration and aggregation phase includes **record linkage** as a specific quality task (see Figure

1). When looking at record linkage for Big Data, there are two major challenges. Big Data technologies can indeed support traditional record linkage methods on data that have a big size: this is indeed the purpose of tools such as Dedoop [134]. Here the challenge is to make traditional algorithms scalable. Second, the record linkage pipeline for Big Data should be fully automatic. This means that several manual-based record linkage steps should be overcome: supervised learning techniques, clerically prepared training sets, user defined classification thresholds, clerical review, interactive evaluation of obtained results [51]. In this respect, in terms of open challenges, fully automated techniques, like e.g. [56], should find more and more space.

#### 4.5 Data Processing and Querying

Storing and querying Big Data sets efficiently are fundamental services of any Big Data platform. The main challenge in this context is to provide scalable and reliable *distributed* storage systems that supports a broad range of data access patterns for different kinds of queries and analytics. Several approaches have been developed towards that end.

*Hadoop ecosystem.* In the context of the Hadoop ecosystem, numerous technologies have been developed to support management and processing of Big Data. A fundamental component is the HDFS distributed file system, which was originally inspired by the Google file system [135]. HDFS aims at: (i) scalability by storing large files across multiple nodes (so called DataNodes) as well as at (ii) reliability by replicating blocks of files across multiple nodes (with default replication factor set to 3). For better read performance, HDFS tries to exploit data locality by executing mapper tasks on the nodes storing the required data. However, HDFS is optimized for large and rather immutable files. In addition to standard text files such as text, CSV, etc. several more advanced file formats are available, including columnar storage formats such as Parquet and ORC, which also support compression, indexing and Bloom filters. Other approaches such as HAIL [136] propose clustered indexes, which are created upon upload of the data to HDFS in order to speed up query processing.

HDFS provides a foundation for several MapReduce-like data processing frameworks such as Hadoop MapReduce, Apache Spark or Flink [137]. The latter two extend the original MapReduce model by additional programming abstractions simplifying the formulation of complex analytical pipelines as well as an in-memory processing model for better performance.

<sup>3</sup> <https://www.wikidata.org>

<sup>4</sup> <http://wiki.dbpedia.org/>

<sup>5</sup> <https://www.google.com/intl/en-419/insidesearch/features/search/knowledge.html>

The core of Spark<sup>6</sup> are abstractions for distributed in-memory and immutable data collections, which can be transformed by parallel operations (including relational query operations). Originally, so-called RDDs (resilient distributed datasets) [138] were used for this purpose: recently so-called DataSets have been introduced, which are strongly-typed and allow a better optimization of (relational) queries [139]. In addition to the batch-processing model, both Spark and Flink also provide extensions for processing data streams.

Another part of the Hadoop ecosystem based on HDFS is HBase, which was inspired by Google’s BigTable approach [76]. HBase is a distributed database implementing a wide-column model. Though, it does not support a declarative query interface directly, several extensions have been developed on top of HBase, such as Drill and Phoenix as SQL layers for HBase, and Hive as a data warehouse solution.

Particularly for Spark and Flink, several higher level frameworks for different data models exist that exploit the underlying parallel dataflow engine for scalable processing. This includes extensions for graph processing and analytics such as GraphX [57] and GRADOOP [58] for processing RDF data using SPARQL BGPs, for spatial data processing, e.g. GeoSpark [59], SpatialSpark [60], and STARK [140], as well as machine learning frameworks such as MLlib [141].

*S\*QL on Hadoop.* SQL-on-Hadoop systems can be seen as a class of database systems running on cluster infrastructures. However, they differ from traditional systems by relying on scalable storage layers and cluster resource management frameworks (such as Hadoop’s YARN or Mesos<sup>7</sup>). In that sense, cluster resources can be shared with other jobs, e.g., batch jobs for preparing data or iterative jobs performing complex machine learning tasks.

One of the first attempts to build a database system using Hadoop is HadoopDB [142], where Hadoop is used as a coordinator to connect multiple single-node databases. Other examples are Hive, HAWQ (formerly known as Pivotal), Impala [62], and VectorH [63]. Particularly, VectorH does not only use HDFS for compressed columnar storage and YARN for workload management, but also exploits HDFS block placement policy for locality and supports transactional updates. A specific feature of Impala is the just-in-time code generator for query compilation. SparkSQL [139] is a distributed query engine integrating relational processing with Spark’s functional programming API. SparkSQL supports both a SQL interface as well as a declarative API that integrates with procedural code. It includes an

extensible optimizer supporting lower-level code generation to speed up query execution.

In addition to SQL support on Hadoop, there exist also engines for other query languages like SPARQL. Examples are HadoopRDF [143], Sempala [144] for translating SPARQL to Impala’s SQL, and S2RDF for translating SPARQL to Spark [64].

Besides Hadoop, numerous data management architectures exist aiming at the management of Big (semantic) Data. The area of NoSQL databases covers a wide range of solutions from scalable key-value stores over document stores to graph databases. Notable examples includes CumulusRDF [145], which is based on Cassandra and implements a SPARQL query interface, Couchbase or MongoDB, which allows to store and query tree/graph structures as JSON documents, distributed graph databases, such as Neo4j and AllegroGraph, for storing and processing large graphs, and scalable triple stores such as Virtuoso or GraphDB providing a SPARQL interface [146].

Data processing and querying can have a quality counterpart in terms of **quality access and querying** (see Figure 1). Quality-driven data access has been investigated in contexts where multiple sources could contribute answering to user queries. For Big Data sources this paradigm can be particularly relevant due to the uncontrolled data generation mechanism that is often inherent to such sources; indeed, even if the data generation is out of control, a user interested in the data can rely on their quality features.

## 4.6 Data Analysis and Modelling

Executing Analytics on Big Data imposes new challenges.

Traditional algorithms for data mining assume to have access to the entire dataset; while high data volumes and real-time processing give access only to fractions of the dataset. For example, data transformation techniques adopted in batch scenarios can be inappropriate in case of data distributions evolving over time [147]. Moreover, data streams, as ordered and potentially unbounded sequences of data points, typically create non-stationary flows, in which the data distribution evolves over time. This means that finite training sets and static models are no longer appropriate [148]. The situation is worsen by the fact that data is distributed over a variety of different sources having diversified latencies.

In Big Data, complex analytical procedures and methods of various fields, e.g., machine learning, data mining, statistics, and mathematics are often com-

<sup>6</sup> <https://spark.apache.org/>

<sup>7</sup> <http://mesos.apache.org/>

bined [66,149], as no single algorithm can perform optimally in all cases. Then, various methods have been proposed to support model-selection, based on the observed data and the analytics goals [37]. This requires the data structure and the semantics of analytics to be expressed in machine readable formats [150,151].

Adaptive Models are also proposed for managing the architecture of large-scale distributed systems [152, 153]. These models provide abstractions of systems during runtime to support dynamic state monitoring. Hartmann et al. [154] go one step further. They combine the idea of runtime models with reactive programming and peer-to-peer distribution. Reactive programming is aimed at supporting interactive applications, which react on events by focusing on streams. For this purpose a typical publish/subscribe pattern is used. Khare et al. show the application of such an approach in the IoT domain in [67]. In [68] semantic models are used to provide a unified view of the heterogeneous elements composing these systems, and reasoning mechanisms are leveraged to drive adaptation strategies.

Even if the primary focus of Big Data analytics do not involve the definition of an end-to-end processes, some authors have studied its application to Business Process [155]. Data science approaches tend to be process agonistic whereas process science approaches tend to be model-driven.

Luckham [156] introduces *Complex Event Processing (CEP)* by defining complex events which are correlated among each other. Saleh et al. [61] apply the data aggregation approach of CEP to data streams. *Process Mining (PM)* is a process-centric management technique bridging the gap between data mining and traditional model-driven *Business Process Management (BPM)* [157,158]. In this field of research, business processes are analyzed on the basis of process execution logs, so-called *event logs*. Events are defined as process steps and event logs as sequential events recorded by an information system [159]. The main objective of PM is to extract valuable, process-related information from logs for providing detailed information about actual processes for analytical purposes, e.g., to identify bottlenecks, to anticipate problems, to record policy violations, to streamline processes, etc. [158]. Current event processing technologies usually monitor only a single stream of events at a time. Even if users monitor multiple streams, they often end up with multiple “silo” views. A more unified view is needed that correlates with events from multiple streams of different sources and in different formats. Thereby, heterogeneity and incompleteness of data are major challenges [81]. Mostly, PM operates on the basis of events that belong to cases that are already completed [160]. This off-line

analysis is not suitable for cases which are still in the pipeline. An open research question is whether current algorithms to abstract models from logs are scalable enough to handle data streams [155,161].

The analysis and modelling phase does also have an important quality counterpart, **quality analysis**, c. f. Figure 1. There are obviously quality metrics depending on specific methods used, e.g. the F-measure for classification tasks [162]. However, methods-independent quality measurements could be considered. In [71], a quality framework for Official Statistics is proposed; an interesting notion is *steady states*, meaning that the data have to be processed through a series of stable representation that can be referenced by future analytics processes.

#### 4.7 Data Interpretation and Reporting

While processing Big Data, a key issue is to support the user’s comprehension of the process result. Visual Analytics is an emerging field of research that aims at combining the automatic computation with visual analysis, allowing the user to combine the two approaches. While this combination has proven to be effective, Big Data poses problems concerning the volume of data to display (asking for more abstract visual representation), the velocity with which the data change, and their variety. Capturing their volume and their variety requires solutions that are able to visually abstract and/or aggregate the data, in order to allow their representation as visual elements in a finite visual space. Capturing data velocity poses additional challenges in terms of how visually convey changes in order to maintain their traceability while ensuring a general stability of the whole visualization, avoiding to confuse the final user. Additionally, in the more demanding scenario of data streams, accuracy of results, trend discovery, and trend anticipation pose challenges for the visualization, like handling uncertainty in the displayed data or visualizing prediction accuracy in trend discovery. In this context, data semantics can provide an additional layer of information to be exploited for mitigating the afore-mentioned issues at different levels (supporting better user interaction, visualizing only the meaningful portions of data, linking semantics with extensional properties of the dataset), Also, visualization can be exploited in order to comprehend and refine the semantic description of the data. In [163] the authors exploit data semantics to automatically generate visualizations. The proposed approach uses two different ontologies, one that maps the semantics of the dataset and another that maps the semantics of a set of visual representations, with the final goal of automatically proposing

the best possible visual representation. On the same topic, [91] extracts, through semantics modeling, a data context used then to recommend a good visual representation paradigm. The works in [164,165] propose a cockpit for ontology visualization, in order to improve the knowledge space exploration, evaluated on Linked Open Data (LOD). Other approaches to ontology visualization and exploration are presented in [166–168]. Data semantics can be useful in the visual analysis process design. Focused on applying a semantic transformation for the taxi trajectories, applied for the Shenzhen city in China, is presented in [169]. It exploits semantic information (street names, points of interest) for discovering hidden patterns in the data and allowing faster and more accurate taxi trajectories visual exploration. Supporting querying of ontology-based Big Data through a visual query language is one of the goals in the Optique project [92,70]. Several efforts have been produced in the application of visualization for representing semantics in social media analysis, recent work [93] proposes a multi-granular, data-driven, and theory-informed framework to interactively explore the pulse of a city based on social media, while other researchers [73] focused on wearable devices data in a health-care domain.

The data interpretation and reporting phase has also a specific quality task in the pipeline of Figure 1. The quality of analyses deriving from Big Data should be carefully evaluated [170]. A relevant example is the Google work on flu trends [74] that estimates flu prevalence from flu-related Internet searches. In January 2013, Google flu trends estimated almost twice as many flu cases as were reported by CDC, the Centers for Disease Control and Prevention. The initial Google paper stated that the Google Flu Trends predictions were 97% accurate compared to CDC data. This case is emblematic of other challenges related to Big Data: (i) evaluation of robustness over time of models based on Big Data that may exhibit unexpected glitches (ii) evaluation of the usage of Big Data-based models alone or in conjunction with more traditional sources.

#### 4.8 Representing Processes

The complexity of Big Data architectures has encouraged the definition of work-flow languages for managing pipelines. Among the most popular solutions we have: Apache Oozie <sup>8</sup>, AirBnB Airflow <sup>9</sup>, LinkedIn Azkaban <sup>10</sup>, and Spring Cloud Data Flow <sup>11</sup>. These frame-

<sup>8</sup> <http://oozie.apache.org>

<sup>9</sup> <https://airflow.apache.org>

<sup>10</sup> <https://azkaban.github.io>

<sup>11</sup> <https://cloud.spring.io/spring-cloud-dataflow/>

works support orchestration of software components written in different languages, enabling the integration of heterogeneous systems and facilitating programmers in choosing their favorite technologies. In principle, these frameworks provide a representation model that can contribute to foster re-usability and modularity. However, the level of portability achieved by these languages is limited [97]. In fact, there is no explicit integration between the execution work-flow and the code that is executed by atomic tasks. This implies that knowledge about task level code is required for interfacing elements. Moreover, these orchestration engines do not provide support for validating a work-flow or for optimization step. In particular, it has been argued that the complex nature of Big Data processing makes optimization strongly context dependent: for example, the effectiveness of a pipeline depends on data distribution and on the parallelization model adopted at the deployment infrastructure.

Recent researches have faced these limitations relay on platform-specific configuration libraries. KeystoneML [98], for example, introduced an approach to large-scale pipeline optimization extending Spark ML libraries [99]. The authors focus on capturing end-to-end pipeline application characteristics that are used to automatically optimize execution at both the operator and pipeline application levels.

A high-level data-flow abstraction for modeling complex pipelines is proposed in [100]. The data-flows proposed in this work are directed acyclic graphs that specify some aspects of a pipeline delegating data inspections and optimization to the execution stage. In [101] the authors proposes an adaptation of TensorFlow, for supporting data analysis, transformation, and validation. The aim is boosting automation in the deployment of machine learning models.

The main limitations of the current proposals are that they are closely tied to specific frameworks, such as Spark in [98,100] or TensorFlow in [101] and lack of a formal definition supporting verification procedures for Big Data pipelines.

Although the above perspectives have been considered in the literature, there is a lack of a comprehensive approach addressing the whole life-cycle of a Big Data campaign. A general methodology for representing and reasoning on all steps of a Big Data pipeline is proposed in [34]. This methodology is used for several applications in the framework of the TOREADOR project<sup>12</sup>.

<sup>12</sup> <http://www.toreador-project.eu>

## 5 Open Challenges

As a result of the discussion proposed in Section 4, we propose a list of challenges that we consider relevant for future research.

- Developing an **integrated data model** (at the conceptual and logical level), capable of representing heterogeneous and complex models of various data sources [90]. These models have to incorporate techniques for **managing data sources**, i.e.,: (semi)-automatic discovery of data sources which are relevant to a user and dynamically plugging-in the data sources into an existing integration architecture. Designing and implementing efficient **Data Integration Architectures** for ingesting data into a DL [171] and further, for producing clean and more structured data. Since a Big Data ETL engine processes much more complex ETL/ELT workflows and much larger data volumes than a standard one, its performance becomes vital. Moreover, to handle the complexity of data, workflows require in-house developed user-defined functions, whose optimization is difficult. Performance optimization of ETL/ELT workflows has not been fully solved for traditional DW architectures and Big Data added new problems into the already existing ones [172]. This includes efficient **mechanisms for storing and retrieving data** in a DL. Finding a relevant data set quickly requires additional data structures (a counterpart of indexes in traditional DBs), physical organization of data (a counterpart of partitioning, row-store, column-store in traditional DBs), and compression algorithms, suitable for complex and heterogeneous data. A first challenge arises from the continuous production of new data combined with the need for real-time or on-line analytics. Thus, Big Data platforms have to cope both with (transient) streaming data and persistent data while being able to process queries on both kinds of data, in the form of continuous queries as well as ad-hoc batch queries.
- Developing a **query language** capable of handling data complexity, heterogeneity, and incompleteness. Moreover, it seems to be important to include user preferences in a query, like quality of service, quality of data, output data format, and preferred way of visualization. Another challenge is the support for declarative queries and their optimization. SQL is often considered as not powerful enough to formulate complex analytical tasks. Therefore, data scientists tend to prefer language-integrated DSLs which basically combines programming languages

with domain specific language, scripting languages like Python, dataflow languages like Pig Latin [173], special-purpose languages like R [174] or implement specific tasks in user-defined functions (UDF). Particularly imperative languages like Python or Java but also black-box UDFs make it difficult to parallelize and optimize complex dataflow programs, although, query optimization is a well-studied field and recent developments, e.g. around Spark and other Hadoop-based systems, show a trend (back) towards declarative and optimizable query languages such as SQL dialects.

- Developing a **metadata standard** and architecture. The latter should support: automatic or semi-automatic metadata discovery and collection from new data sources plugged into an integration system, as well as efficient metadata storing, searching, and visualizing. The benefit of metadata management within Big Data technologies was also established by surveys with Data Science professionals [175,102]. Metadata management opens challenges that affect almost all aspects of Big Data [26]. For example, the data processing engine has to identify the data sets that can be used for starting the ingestion procedure, as there may exist multiple data sets storing semantically the same data. The problem is then to figure out which data sets to use based on the query to be answered or the analytics to be applied. Different criteria may be taken into consideration, e.g., data format, data quality, data completeness, data freshness. Finally, the results must be appropriately visualized [123]. Some work on this issue was initiated in [36,82]. For example, data in various formats could be converted on-the-fly to the format preferred by a user, e.g., relational, XML, graph, RDF. Also, different data sources may have a different impact on performance. Identifying these implications is crucial to avoid overloading a single job.
- Developing solid techniques for dealing with **incomplete and uncertain data**. For analytical purposes (i) events have to be captured from data streams, (ii) events of interest have to be separated from noise, (iii) correlations with other streams and databases have to be established, (iv) reaction to events of interest must happen in real-time, and (v) these events have to be stored in an appropriate model structure equipped to deal with concept drifts detection to then run on-line or off-line analysis. Improving the scalability of probabilistic and uncertainty data models is an important

issue as well as the expressiveness of the data and uncertainty they can manage. Note that there are multiple models for representing uncertainty: for instance, the possibilistic or fuzzy set model [176], and the Dempster-Shafer evidence model [177]. Furthermore, there are many different kinds of integration and data quality problems that require to manage uncertainty. For example, [178] presents an approach for probabilistic integration of data on groupings. It furthermore shows that probabilistic database technology (in this case MayBMS [179]) is already close to being able to handle real-world biological databases. When combined with an effective method for data quality measurement, this technology can deliver a good-enough approach where small iterations reduce the overall effort in data quality improvement.

- Designing and implementing efficient **virtual data integration** architectures, as complementary to a DL or polystore. Such architectures expose their pitfall of being slow, since query resolving and data integration is executed on the fly. For this reason, new optimization techniques are needed. Some of them could be based on caching the results at two levels: in main memory and on disk. Using the cached data requires their management, i.e., to decide what to cache, which queries should be executed on data sources and which on cached data, proactive refreshing is also needed in the spirit of [180]. We can envision additional challenges that cope with the capability to exploit the data semantics to steer the visual analytics process in Big Data analysis. More in detail, data semantics could be exploited as a steering factor in Big Data explorative analysis, following Progressive Visual Analytics techniques [181–184], a novel approach producing intermediate approximated results allows for fast Big Data exploration. In this approach, the user can steer the visual analysis process, and the availability of data semantics can be a way to express steering preferences (e.g., focus the computation only on data having a particular semantics) constitutes a challenge and opportunity. Data semantics can also help while selecting the right visual representations for a dataset, taking into account additional semantic information like, for example, the user’s task and the device capabilities, encouraging the creation of a taxonomy that binds together the semantics and the structure of the data with the appropriate visualization paradigms and techniques. A last challenge is to use visual analytics on Big Data in order to

extract the semantics itself, with a semi-automated process in which the user projects her knowledge of the problem on the data representation (see, e.g., [185] for network visualizations), on a portion of interest of the data or the full dataset. The visual representation of a dataset can help in identifying common properties of the data, trends, features, that all together can help to form a semantic description of the data.

- Developing models to **support reproducibility and verifiability of Big Data pipelines**. *Reproducibility* is a precondition to an efficient link between research and production environments and to support reuse and modularity. It involves the definition of the Extract, Transform, and Load (ETL) process executed, including the data sources integrated with their metadata about provenance and format. *Verifiability* is of fundamental importance because low quality data will necessarily generate low-quality analytics. It involves the definition of input and output data type for each integrated task or methods to examine data distribution in order to verify essential preconditions for statistical analysis. However, achieving these objectives in Big Data architectures is not trivial. It has been acknowledged that implementing complex pipelines for real-world systems poses a huge challenge [75], especially because the effectiveness of a pipeline strictly depends on data distribution. This calls for a representation of the interdependences between the different stages of a pipeline.
- Developing models to represent **regulatory knowledge for automated compliance**. Regulatory peculiarities cannot be addressed on a project-by-project basis. Rather, certified compliance of each Big Data project (e.g., in the form of a Privacy Impact Analysis) should be made available from the outset to all actors that use Big Data analytics in their business model. Also, data processing comes with legal issues that may trigger unwanted litigation. How to account intellectual property and how to shape the economical exploitation of analytics in multi-party environments [186]? How to provide evidence that data processing is compliant with ethics, beyond norms and directives [187]? Those are among the questions that still require mature and reliable solutions.

## 6 Conclusions

The complexity of Big Data applications in conjunction with the lack of standards for representing their components, computations, and processes, have made the design of data-intensive applications a failure prone and resource-intensive activity. In this paper, we argued that no innovation in algorithms can compensate lack of sound modeling practices. Indeed, we believe that the major challenges facing Big Data research require - even more than developing new analytics - devising innovative data management techniques capable to deliver non-functional properties like data quality, data integration, model compliance, or regulatory compliance. Data Semantics research can address such challenges in future research according to the FAIR principles [18], for implementing design procedures that generate *Findable, Accessible, Interoperable, and Reusable* data.

## Acknowledgements

This research was partially supported by the European Union's Horizon 2020 research and innovation programme under the TOREADOR project, grant agreement No 688797. The work of R. Wrembel is supported from the National Science Center grant No. 2015/19/B/ST6/02637.

## References

1. P. Zikopoulos, C. Eaton *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.
2. J. S. Ward and A. Barker, "Undefined by data: a survey of big data definitions," *arXiv preprint arXiv:1309.5821*, 2013.
3. M. A. Beyer and D. Laney, "The importance of big data: a definition," *Stamford, CT: Gartner*, pp. 2014–2018, 2012.
4. D. Laney, "3d data management: Controlling data volume, velocity and variety," *META Group Research Note*, vol. 6, p. 70, 2001.
5. I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of ?big data? on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.
6. S. F. Wamba, S. Akter, A. Edwards, G. Chopin, and D. Gnanzou, "How big data can make big impact: Findings from a systematic review and a longitudinal case study," *International Journal of Production Economics*, vol. 165, pp. 234 – 246, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925527314004253>
7. S. Madden, "From databases to big data," *IEEE Internet Computing*, vol. 16, no. 3, pp. 4–6, 2012.
8. A. Amazon, "Amazon," 2016. [Online]. Available: [2016-01-06]<https://aws.amazon.com>
9. A. Hadoop, "Hadoop," 2009. [Online]. Available: [2009-03-06]<http://hadoop.apache.org>
10. H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact." *MIS quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012.
11. X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *ieee transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2014.
12. M. Hilbert, "Big data for development: A review of promises and challenges," *Development Policy Review*, vol. 34, no. 1, pp. 135–174, 2016.
13. M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya, "Big data computing and clouds: Trends and future directions," *Journal of Parallel and Distributed Computing*, vol. 79, pp. 3–15, 2015.
14. V. Markl, "Breaking the chains: On declarative data analysis and data independence in the big data era," *Proc. of VLDB Endowment*, vol. 7, no. 13, pp. 1730–1733, August 2014.
15. E. Damiani, B. Oliboni, E. Quintarelli, and L. Tanca, "Modeling semistructured data by using graph-based constraints," in *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*. Springer, 2003, pp. 20–21.
16. J. Poole, D. Chang, D. Tolbert, and D. Mellor, *Common Warehouse Metamodel. Developer's Guide*. Wiley, 2003.
17. C. Ardagna, R. Asal, E. Damiani, and Q. Vu, "From security to assurance in the cloud: A survey," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, pp. 2:1–2:50, August 2015.
18. M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, p. 160018, 2016.
19. K. Aberer, T. Catarci, P. Cudré-Mauroux, T. Dillon, S. Grimm, M.-S. Hacid, A. Illarramendi, M. Jarrar, V. Kashyap, M. Mecella *et al.*, "Emergent semantics systems," in *Semantics of a Networked World. Semantics for Grid Databases*. Springer, 2004, pp. 14–43.
20. P. Cudré-Mauroux, K. Aberer, A. I. Abdelmoty, T. Catarci, E. Damiani, A. Illaramendi, M. Jarrar, R. Meersman, E. J. Neuhold, C. Parent *et al.*, "Viewpoints on emergent semantics," in *Journal on Data Semantics VI*. Springer, 2006, pp. 1–27.
21. C. A. Ardagna, P. Ceravolo, and E. Damiani, "Big data analytics as-a-service: Issues and challenges," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3638–3644.
22. M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.
23. A. Azzini and P. Ceravolo, "Consistent process mining over big data triple stores," in *Big Data (BigData Congress), 2013 IEEE International Congress on*. IEEE, 2013, pp. 54–61.
24. J. Wood, "What's in a link," *Readings in Knowledge Representation*. Morgan Kaufmann, 1985.
25. M. J. Franklin, A. Y. Halevy, and D. Maier, "From databases to dataspace: a new abstraction for information management," *SIGMOD Record*, vol. 34, no. 4, pp. 27–33, 2005. [Online]. Available: <http://doi.acm.org/10.1145/1107499.1107502>
26. K. Smith, L. Seligman, A. Rosenthal, C. Kurcz, M. Greer, C. Macheret, M. Sexton, and A. Eckstein, "'big metadata': The need for principled metadata management in big data ecosystems," in *Proceedings of Workshop on Data Analytics in the Cloud*, ser.



- DanaC'14. New York, NY, USA: ACM, 2014, pp. 13:1–13:4. [Online]. Available: <http://doi.acm.org/10.1145/2627770.2627776>
27. M. A. Waller and S. E. Fawcett, “Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management,” *Journal of Business Logistics*, vol. 34, no. 2, pp. 77–84, 2013.
  28. V. Borkar, M. J. Carey, and C. Li, “Inside big data management: ogres, onions, or parfaits?” in *Proceedings of the 15th international conference on extending database technology*. ACM, 2012, pp. 3–14.
  29. T. White, *Hadoop: The definitive guide*. ” O’Reilly Media, Inc.”, 2012.
  30. H. Jagadish, “Big data and science: myths and reality,” *Big Data Research*, vol. 2, no. 2, pp. 49–52, 2015.
  31. P. Pääkkönen and D. Pakkala, “Reference architecture and classification of technologies, products and services for big data systems,” *Big Data Research*, vol. 2, no. 4, pp. 166–186, 2015.
  32. C. Ardagna, V. Bellandi, M. Bezzi, P. Ceravolo, E. Damiani, and C. Hebert, “A model-driven methodology for big data analytics-as-a-service,” in *Proc. of Big-Data Congress 2017*, Honolulu, HI, USA, June 2017.
  33. “Challenges and opportunities with big data,” 2012, available on line. [Online]. Available: <https://www.purdue.edu/discoverypark/cyber/assets/pdfs/BigDataWhitePaper.pdf>
  34. C. A. Ardagna, V. Bellandi, M. Bezzi, P. Ceravolo, E. Damiani, and C. Hebert, “Model-based big data analytics-as-a-service: Take big data to the next level,” *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1–1, 2018.
  35. C. Liao and A. Squicciarini, “Towards provenance-based anomaly detection in mapreduce,” in *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*. IEEE, 2015, pp. 647–656.
  36. J. Duggan, A. J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner, S. Madden, D. Maier, T. Mattson, and S. Zdonik, “The BigDAWG Polystore System,” *SIGMOD Record*, vol. 44, no. 2, 2015.
  37. R. Sowmya and K. Suneetha, “Data mining with big data,” in *Intelligent Systems and Control (ISCO), 2017 11th International Conference on*. IEEE, 2017, pp. 246–250.
  38. W. Zhou, S. Mapara, Y. Ren, Y. Li, A. Haeberlen, Z. Ives, B. T. Loo, and M. Sherr, “Distributed time-aware provenance,” in *Proceedings of the VLDB Endowment*, vol. 6, no. 2. VLDB Endowment, 2012, pp. 49–60.
  39. S. Akoush, R. Sohan, and A. Hopper, “Hadoopprov: Towards provenance as a first class citizen in mapreduce,” in *TaPP*, 2013.
  40. B. Glavic, “Big data provenance: Challenges and implications for benchmarking,” in *Specifying big data benchmarks*. Springer, 2014, pp. 72–80.
  41. L. Berti-Equille and M. L. Ba, “Veracity of big data: Challenges of cross-modal truth discovery,” *J. Data and Information Quality*, vol. 7, no. 3, pp. 12:1–12:3, 2016.
  42. M. Kläs, W. Putz, and T. Lutz, “Quality evaluation for big data: A scalable assessment approach and first evaluation results,” in *Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2016 Joint Conference of the International Workshop on*. IEEE, 2016, pp. 115–124.
  43. J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes, “Improving efficiency and accuracy in multilingual entity extraction,” in *Proceedings of the 9th International Conference on Semantic Systems*. ACM, 2013, pp. 121–124.
  44. J. Shin, S. Wu, F. Wang, C. De Sa, C. Zhang, and C. Ré, “Incremental knowledge base construction using DeepDive, journal = Proc. VLDB Endowment, volume = 8, number = 11, month = jul, year = 2015, issn = 2150-8097, pages = 1310–1321, doi = 10.14778/2809974.2809991, note = DOI 10.14778/2809974.2809991.”
  45. L. Chiticariu, R. Krishnamurthy, Y. Li, S. Raghavan, F. R. Reiss, and S. Vaithyanathan, “Systemt: an algebraic approach to declarative information extraction,” in *Proceedings of the Association for Computational Linguistics*, 2010, pp. 128–137.
  46. P. Fuhring and F. Naumann, “Emergent data quality annotation and visualization,” 2007, available on line. Date accessed: 2018. [Online]. Available: [https://hpi.de/fileadmin/user\\_upload/fachgebiete/naumann/publications/2007/Emergent\\_Data\\_Quality\\_Annotation\\_and\\_Visualization.pdf](https://hpi.de/fileadmin/user_upload/fachgebiete/naumann/publications/2007/Emergent_Data_Quality_Annotation_and_Visualization.pdf)
  47. C. Bondiombouy, B. Kolev, O. Levchenko, and P. Valduriez, “Multistore big data integration with cloudmssql,” in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXVIII*. Springer, 2016, pp. 48–74.
  48. S. Bergamaschi, D. Beneventano, F. Mandreoli, R. Martoglia, F. Guerra, M. Orsini, L. Po, M. Vincini, G. Simonini, S. Zhu *et al.*, “From data integration to big data integration,” in *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer, 2018, pp. 43–59.
  49. R. Ramakrishnan, B. Sridharan, J. R. Douceur, P. Kasturi, B. Krishnamachari-Sampath, K. Krishnamoorthy, P. Li, M. Manu, S. Michaylov, R. Ramos *et al.*, “Azure data lake store: A hyperscale distributed file service for big data analytics,” in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 51–63.
  50. M. Masseroli, A. Kaitoua, P. Pinoli, and S. Ceri, “Modeling and interoperability of heterogeneous genomic big data for integrative processing and querying,” *Methods*, vol. 111, pp. 3–11, 2016.
  51. M. Scannapieco, A. Virgillito, and D. Zardetto, “Placing big data in official statistics: A big challenge?” in *Proceedings of NTTS (New Techniques and Technologies for Statistics), March 5-7, 2013, Brussels*, 2013.
  52. M. Gualtieri and B. Hopkins, “SQL-For-Hadoop: 14 Capable Solutions Reviewed,” 2014, forrester.
  53. H. Liu, T. A. Kumar, and J. P. Thomas, “Cleaning framework for big data-object identification and linkage,” in *Big Data (BigData Congress), 2015 IEEE International Congress on*. IEEE, 2015, pp. 215–221.
  54. M. A. Gulzar, M. Interlandi, X. Han, M. Li, T. Condie, and M. Kim, “Automated debugging in data-intensive scalable computing,” in *Proceedings of the 2017 Symposium on Cloud Computing*, ser. SoCC ’17. New York, NY, USA: ACM, 2017, pp. 520–534. [Online]. Available: <http://doi.acm.org/10.1145/3127479.3131624>
  55. T. de Wit, “Using ais to make maritime statistics,” in *Proceedings of NTTS (New Techniques and Technologies for Statistics), March 14-16, 2017, Brussels*, 2017.
  56. D. Zardetto, M. Scannapieco, and T. Catarci, “Effective automated object matching,” in *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*, 2010, pp. 757–768.

57. R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica, "Graphx: a resilient distributed graph system on spark," in *First International Workshop on Graph Data Management Experiences and Systems, GRADES 2013, co-located with SIGMOD/PODS 2013, New York, NY, USA, June 24, 2013*, 2013, p. 2. [Online]. Available: <http://event.cwi.nl/grades2013/02-xin.pdf>
58. M. Junghanns, A. Petermann, K. Gómez, and E. Rahm, "GRADOOP: scalable graph data management and analytics with hadoop," *CoRR*, vol. abs/1506.00548, 2015. [Online]. Available: <http://arxiv.org/abs/1506.00548>
59. J. Yu, J. Wu, and M. Sarwat, "Geospark: a cluster computing framework for processing large-scale spatial data," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Bellevue, WA, USA, November 3-6, 2015*, 2015, pp. 70:1–70:4. [Online]. Available: <http://doi.acm.org/10.1145/2820783.2820860>
60. S. You, J. Zhang, and L. Gruenwald, "Large-scale spatial join query processing in cloud," in *31st IEEE International Conference on Data Engineering Workshops, ICDE Workshops, Seoul, South Korea, April 13-17, 2015*, 2015, pp. 34–41. [Online]. Available: <http://dx.doi.org/10.1109/ICDEW.2015.7129541>
61. O. Saleh, S. Hagedorn, and K. Sattler, "Complex event processing on linked stream data," *Datenbank-Spektrum*, vol. 15, no. 2, pp. 119–129, 2015.
62. M. Kornacker, A. Behm, V. Bittorf, T. Bobrovitsky, C. Ching, A. Choi, J. Erickson, M. Grund, D. Hecht, M. Jacobs, I. Joshi, L. Kuff, D. Kumar, A. Leblang, N. Li, I. Pandis, H. Robinson, D. Rorke, S. Rus, J. Russell, D. Tsirogiannis, S. Wanderman-Milne, and M. Yoder, "Impala: A modern, open-source SQL engine for hadoop," in *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*, 2015. [Online]. Available: [http://www.cidrdb.org/cidr2015/Papers/CIDR15\\_Paper28.pdf](http://www.cidrdb.org/cidr2015/Papers/CIDR15_Paper28.pdf)
63. A. Costea, A. Ionescu, B. Raducanu, M. Switakowski, C. Bârca, J. Sompolski, A. Luszczak, M. Szafranski, G. de Nijs, and P. A. Boncz, "Vectorh: Taking sql-on-hadoop to the next level," in *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, 2016, pp. 1105–1117. [Online]. Available: <http://doi.acm.org/10.1145/2882903.2903742>
64. A. Schätzle, M. Przyjacieli-Zablocki, S. Skilevic, and G. Lausen, "S2RDF: RDF querying with SPARQL on spark," *PVLDB*, vol. 9, no. 10, pp. 804–815, 2016. [Online]. Available: <http://www.vldb.org/pvldb/vol9/p804-schaetzle.pdf>
65. P. Cudré-Mauroux, I. Enchev, S. Fundatureanu, P. T. Groth, A. Haque, A. Harth, F. L. Keppmann, D. P. Miranker, J. Sequeda, and M. Wylot, "Nosql databases for RDF: an empirical evaluation," in *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, 2013, pp. 310–325. [Online]. Available: <http://dx.doi.org/10.1007/978-3-642-41338-4.20>
66. A. Appice, M. Ceci, and D. Malerba, "Relational data mining in the era of big data," in *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer, 2018, pp. 323–339.
67. S. Khare, K. An, A. S. Gokhale, S. Tambe, and A. Meena, "Reactive stream processing for data-centric publish/subscribe," in *Proceedings of the 9th International Conference on Distributed Event-Based Systems (DEBS)*. ACM, 2015, pp. 234–245.
68. F. Poggi, D. Rossi, P. Ciancarini, and L. Bompani, "Semantic run-time models for self-adaptive systems: A case study," in *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2016 IEEE 25th International Conference on*. IEEE, 2016, pp. 50–55.
69. J.-H. Um, S. Lee, T.-H. Kim, C.-H. Jeong, S.-K. Song, and H. Jung, "Semantic complex event processing model for reasoning research activities," *Neurocomputing*, vol. 209, pp. 39–45, 2016.
70. M. Giese, A. Soyulu, G. Vega-Gorgojo, A. Waaler, P. Haase, E. Jiménez-Ruiz, D. Lanti, M. Rezk, G. Xiao, Ö. Özgep *et al.*, "Optique: Zooming in on big data," *Computer*, vol. 48, no. 3, pp. 60–67, 2015.
71. "Unece big data quality framework," available online. [Online]. Available: <http://www1.unece.org/stat/platform/display/bigdata/2014+Project>
72. J. Severin, M. Lizio, J. Harshbarger, H. Kawaji, C. O. Daub, Y. Hayashizaki, N. Bertin, A. R. Forrest, F. Consortium *et al.*, "Interactive visualization and analysis of large-scale sequencing datasets using zenbu," *Nature biotechnology*, vol. 32, no. 3, pp. 217–219, 2014.
73. E. Mezghani, E. Exposito, K. Drira, M. Da Silveira, and C. Pruski, "A semantic big data platform for integrating heterogeneous wearable data in healthcare," *Journal of medical systems*, vol. 39, no. 12, p. 185, 2015.
74. J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, no. 7232, pp. 1012–1014, 2009.
75. D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems*, 2015, pp. 2503–2511.
76. F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 4:1–4:26, 2008. [Online]. Available: <http://doi.acm.org/10.1145/1365815.1365816>
77. I. Suriarachchi and B. Plale, "Provenance As Essential Infrastructure for Data Lakes," in *Proc. of Int. Work. on Provenance and Annotation of Data and Processes*. LNCS 9672, 2016.
78. I. Terrizzano, P. Schwarz, M. Roth, and J. E. Colino, "Data Wrangling: The Challenging Journey from the Wild to the Lake," in *Proc. of Conf. on Innovative Data Systems Research (CIDR)*, 2015.
79. Teradata, "Putting the data lake to work: A guide to best practices," 2014, accessed on 20 Jun, 2017. [Online]. Available: <http://www.teradata.com/Resources/Best-Practice-Guides/Putting-the-Data-Lake-to-Work-A-Guide-to-Bes>
80. C. Batini and M. Scannapieco, *Data and Information Quality - Dimensions, Principles and Techniques*, ser. Data-Centric Systems and Applications. Springer, 2016.
81. D. Agrawal, P. Bernstein, E. Bertino, S. Davidson, U. Dayal, M. Franklin, J. Gehrke, L. Haas, A. Halevy, J. Han *et al.*, "Challenges and opportunities with big data," Purdue University, Cyber Center Technical Reports, Tech. Rep., 2011.

82. M. Liu and Q. Wang, "Rogas: A declarative framework for network analytics," *Proc. of Int. Conf. on Very Large Data Bases (VLDB)*, vol. 9, no. 13, pp. 1561–1564, 2016.
83. O. Hasan, B. Habegger, L. Brunie, N. Bennani, and E. Damiani, "A discussion of privacy challenges in user profiling with big data techniques: The eexcess use case," in *Big Data (BigData Congress), 2013 IEEE International Congress on.* IEEE, 2013, pp. 25–30.
84. A. Doan, A. Ardalani, J. R. Ballard, S. Das, Y. Govind, P. Konda, H. Li, E. Paulson, H. Zhang *et al.*, "Toward a system building agenda for data integration," *arXiv preprint arXiv:1710.00027*, 2017.
85. M. Flood, J. Grant, H. Luo, L. Raschid, I. Soboroff, and K. Yoo, "Financial entity identification and information integration (feiii) challenge: The report of the organizing committee," in *Proceedings of the Second International Workshop on Data Science for Macro-Modeling.* ACM, 2016, p. 1.
86. A. F. Haryadi, J. Hulstijn, A. Wahyudi, H. Van Der Voort, and M. Janssen, "Antecedents of big data quality: An empirical examination in financial service organizations," in *Big Data (Big Data), 2016 IEEE International Conference on.* IEEE, 2016, pp. 116–121.
87. F. Benedetti, D. Beneventano, and S. Bergamaschi, *Context Semantic Analysis: A Knowledge-Based Technique for Computing Inter-document Similarity.* Springer International Publishing, 2016, pp. 164–178.
88. E. Ford, J. A. Carroll, H. Smith, D. Scott, and J. Caspell, "Extracting information from the text of electronic medical records to improve case detection: A systematic review," vol. 23, 02 2016.
89. D. Haas, S. Krishnan, J. Wang, M. J. Franklin, and E. Wu, "Wisteria: Nurturing scalable data cleaning infrastructure," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 2004–2007, Aug. 2015, doi 10.14778/2824032.2824122.
90. J. Cabot, D. Toman, J. Parsons, O. Pastor, and R. Wrembel, "Big data and conceptual models: Are they mutually compatible?" in *Int. Conf. on Conceptual Modeling (ER)*, 2016, panel discussion. [Online]. Available: <http://er2016.cs.titech.ac.jp/program/panel.html>
91. M. Voigt, S. Pietschmann, L. Grammel, and K. Meißner, "Context-aware recommendation of visualization components," in *Proceedings of the 4th International Conference on Information, Process, and Knowledge Management.* Citeseer, 2012, pp. 101–109.
92. A. Soyly, M. Giese, E. Jimenez-Ruiz, E. Kharlamov, D. Zheleznyakov, and I. Horrocks, "Optiquevqs: Towards an ontology-based visual query system for big data," in *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems*, ser. MEDES '13. New York, NY, USA: ACM, 2013, pp. 119–126, date accessed: 2018. [Online]. Available: <http://doi.acm.org/10.1145/2536146.2536149>
93. G. McKenzie, K. Janowicz, S. Gao, J.-A. Yang, and Y. Hu, "Poi pulse: A multi-granular, semantic signature-based information observatory for the interactive visualization of big geosocial data," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 50, no. 2, pp. 71–85, 2015.
94. M. Habib and M. Van Keulen, "Twitterneed: A hybrid approach for named entity extraction and disambiguation for tweet," *Natural Language Engineering*, vol. FirstView, pp. 1–34, 7 2015, doi 10.1017/S1351324915000194.
95. M. Magnani and D. Montesi, "A survey on uncertainty management in data integration," *JDIQ*, vol. 2, no. 1, pp. 5:1–5:33, 2010, doi 10.1145/1805286.1805291.
96. M. van Keulen, "Managing uncertainty: The road towards better data interoperability," *IT - Information Technology*, vol. 54, no. 3, pp. 138–146, May 2012, doi 10.1524/itit.2012.0674.
97. P. Andrews, A. Kalro, H. Mehanna, and A. Sidorov, "Productionizing machine learning pipelines at scale," in *Machine Learning Systems workshop at ICML*, 2016.
98. E. R. Sparks, S. Venkataraman, T. Kaftan, M. J. Franklin, and B. Recht, "Keystone1: Optimizing pipelines for large-scale advanced analytics," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, April 2017, pp. 535–546.
99. X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen *et al.*, "Mllib: Machine learning in apache spark," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1235–1241, 2016.
100. J.-H. Böse, V. Flunkert, J. Gasthaus, T. Januschowski, D. Lange, D. Salinas, S. Schelter, M. Seeger, and Y. Wang, "Probabilistic demand forecasting at scale," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1694–1705, 2017.
101. D. Baylor, E. Breck, H.-T. Cheng, N. Fiedel, C. Y. Foo, Z. Haque, S. Haykal, M. Ispir, V. Jain, L. Koc *et al.*, "Tfx: A tensorflow-based production-scale machine learning platform," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2017, pp. 1387–1395.
102. C. Ardagna, P. Ceravolo, G. L. Cota, M. M. Kiani, and E. Damiani, "What are my users looking for when preparing a big data campaign," in *Big Data (BigData Congress), 2017 IEEE International Congress on.* IEEE, 2017, pp. 201–208.
103. C. Palmér, "Modelling eu directive 2016/680 using enterprise architecture," 2017.
104. M. Atzmueller, P. Kluegl, and F. Puppe, "Rule-based information extraction for structured data acquisition using textmarker." in *Proceedings of LWA*, 2008, pp. 1–7.
105. B. Settles, "Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances," in *Proceedings of EMNLP.* ACL, 2011, pp. 1467–1478.
106. C. Müller and M. Strube, "Multi-level annotation of linguistic data with MMAX2," *Corpus technology and language pedagogy: New resources, new tools, new methods*, vol. 3, pp. 197–214, 2006.
107. P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "Brat: a web-based tool for nlp-assisted text annotation," in *European Chapter of the Association for Computational Linguistics Demonstrations.* Association for Computational Linguistics, 2012, pp. 102–107.
108. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, *DBpedia: A Nucleus for a Web of Open Data.*
109. C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, no. 3, pp. 1–22, 2009.
110. D. Benikova and C. Biemann, "Semreldata ? multilingual contextual annotation of semantic relations between nominals: Dataset and guidelines," in *LREC*, 2016.
111. A. Lu, W. Wang, M. Bansal, K. Gimpel, and K. Livescu, "Deep multilingual correlation for improved word embeddings," in *HLT-NAACL*, 2015.

112. P. Pecina, A. Toral, A. Way, V. Papavassiliou, P. Prokopoulos, and M. Giagkou, "Towards using web-crawled data for domain adaptation in statistical machine translation," in *The 15th conference of the European Association for Machine Translation (EAMT)*, May 2011.
113. T. Yasseri, A. Spierri, M. Graham, and J. Kertesz, "The most controversial topics in wikipedia: A multilingual and geographical analysis," *Global Wikipedia: International and Cross-Cultural Issues in Online Collaboration*, May 2014.
114. J. C. Micher, "Improving domain-specific machine translation by constraining the language model," Army Research Laboratory, Tech. Rep. ARL-TN-0492, July 2012.
115. J. D?Haen, D. V. den Poel, D. Thorleuchter, and D. Benoit, "Integrating expert knowledge and multilingual web crawling data in a lead qualification system," *Decision Support Systems*, vol. 82, pp. 69 – 78, 2016.
116. M. A. Helou, M. Palmonari, and M. Jarrar, "Effectiveness of automatic translations for cross-lingual ontology mapping," *J. Artif. Int. Res.*, vol. 55, no. 1, pp. 165–208, Jan. 2016.
117. D. Furno, V. Loia, M. Veniero, M. Anisetti, V. Bellandi, P. Ceravolo, and E. Damiani, "Towards an agent-based architecture for managing uncertainty in situation awareness," in *Intelligent Agent (IA), 2011 IEEE Symposium On*. IEEE, 2011, pp. 1–6.
118. N. Dalvi, C. Ré, and D. Suciu, "Probabilistic databases: Diamonds in the dirt," *Communications of the ACM*, vol. 52, no. 7, pp. 86–94, Jul. 2009, DOI 10.1145/1538788.1538810.
119. P. Ceravolo, E. Damiani, and C. Fugazza, "Trustworthiness-related uncertainty of semantic web-style metadata: A possibilistic approach," in *ISWC Workshop on Uncertainty Reasoning for the Semantic Web (URSW)*, vol. 327. [Sn], 2007, pp. 131–132.
120. F. Panse, M. van Keulen, and N. Ritter, "Indeterministic handling of uncertain decisions in deduplication," *JDIQ*, vol. 4, no. 2, pp. 9:1–9:25, Mar. 2013, DOI 10.1145/2435221.2435225.
121. Z. Abedjan, L. Golab, and F. Naumann, "Profiling relational data: A survey," *The VLDB Journal*, vol. 24, no. 4, pp. 557–581, Aug. 2015, DOI 10.1007/s00778-015-0389-y.
122. T. Papenbrock, J. Ehrlich, J. Marten, T. Neubert, J.-P. Rudolph, M. Schönberg, J. Zwiener, and F. Naumann, "Functional dependency discovery: An experimental evaluation of seven algorithms," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 1082–1093, 2015.
123. C. L. P. Chen and C. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Information Sciences*, vol. 275, pp. 314–347, 2014.
124. F. Naumann, "Data profiling revisited," *SIGMOD Record*, vol. 42, no. 4, pp. 40–49, 2014.
125. A. Ahmadov, M. Thiele, J. Eberius, W. Lehner, and R. Wrembel, "Towards a hybrid imputation approach using web tables," in *IEEE/ACM Int. Symp. on Big Data Computing (BDC)*, 2015, pp. 21–30.
126. A. Ahmadov, M. Thiele, W. Lehner, and R. Wrembel, "Context similarity for retrieval-based imputation," in *Int. Symp. on Foundations and Applications of Big Data Analytics (FAB)*, 2017, to appear.
127. Z. Li, M. A. Sharaf, L. Sitbon, S. Sadiq, M. Indulska, and X. Zhou, "A web-based approach to data imputation," *World Wide Web*, vol. 17, no. 5, pp. 873–897, 2014.
128. X. Miao, Y. Gao, S. Guo, and W. Liu, "Incomplete data management: a survey," *Frontiers of Computer Science*, pp. 1–22, 2017.
129. G. Wiederhold, "Mediators in the architecture of future information systems," *IEEE Computer*, vol. 25, no. 3, 1992.
130. A. Tonon, G. Demartini, and P. Cudré-Mauroux, "Combining inverted indices and structured search for ad-hoc object retrieval," in *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, 2012, pp. 125–134. [Online]. Available: <http://doi.acm.org/10.1145/2348283.2348304>
131. M. Catasta, A. Tonon, G. Demartini, J. Ranvier, K. Aberer, and P. Cudré-Mauroux, "B-hist: Entity-centric search over personal web browsing history," *J. Web Sem.*, vol. 27, pp. 19–25, 2014. [Online]. Available: <https://doi.org/10.1016/j.websem.2014.07.003>
132. M. D. Flood, H. Jagadish, L. Raschid *et al.*, "Big data challenges and opportunities in financial stability monitoring," *Banque de France, Financial Stability Review*, vol. 20, 2016.
133. L. M. Ni, H. Tan, and J. Xiao, "Rethinking big data in a networked world," *Frontiers of Computer Science*, vol. 10, no. 6, pp. 965–967, 2016.
134. L. Kolb, A. Thor, and E. Rahm, "Dedoop: Efficient deduplication with hadoop," *PVLDB*, vol. 5, no. 12, pp. 1878–1881, 2012.
135. S. Ghemawat, H. Gobioff, and S. Leung, "The google file system," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles 2003, SOSP 2003, Bolton Landing, NY, USA, October 19-22, 2003*, 2003, pp. 29–43. [Online]. Available: <http://doi.acm.org/10.1145/945445.945450>
136. J. Dittrich, J. Quiané-Ruiz, S. Richter, S. Schuh, A. Jindal, and J. Schad, "Only aggressive elephants are fast elephants," *PVLDB*, vol. 5, no. 11, pp. 1591–1602, 2012. [Online]. Available: [http://vldb.org/pvldb/vol5/p1591\\_jensdittrich\\_vldb2012.pdf](http://vldb.org/pvldb/vol5/p1591_jensdittrich_vldb2012.pdf)
137. P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, "Apache flink™: Stream and batch processing in a single engine," *IEEE Data Eng. Bull.*, vol. 38, no. 4, pp. 28–38, 2015. [Online]. Available: <http://sites.computer.org/debull/A15dec/p28.pdf>
138. M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauly, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012, San Jose, CA, USA, April 25-27, 2012*, 2012, pp. 15–28. [Online]. Available: <https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/zaharia>
139. M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, and M. Zaharia, "Spark SQL: relational data processing in spark," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, 2015, pp. 1383–1394. [Online]. Available: <http://doi.acm.org/10.1145/2723372.2742797>
140. S. Hagedorn, P. Götze, and K. Sattler, "The STARK framework for spatio-temporal data analytics on spark," in *Datenbanksysteme für Business, Technologie und Web (BTW 2017), 17. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS), 6.-10. März*

- 2017, Stuttgart, Germany, Proceedings, 2017, pp. 123–142.
141. X. Meng, J. K. Bradley, B. Yavuz, E. R. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. B. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, “Mllib: Machine learning in apache spark,” *Journal of Machine Learning Research*, vol. 17, pp. 34:1–34:7, 2016. [Online]. Available: <http://jmlr.org/papers/v17/15-237.html>
  142. A. Abouzeid, K. Bajda-Pawlikowski, D. J. Abadi, A. Rasin, and A. Silberschatz, “Hadoopdb: An architectural hybrid of mapreduce and DBMS technologies for analytical workloads,” *PVLDB*, vol. 2, no. 1, pp. 922–933, 2009. [Online]. Available: <http://www.vldb.org/pvldb/2/vldb09-861.pdf>
  143. J. Du, H. Wang, Y. Ni, and Y. Yu, “Hadooprdf: A scalable semantic data analytical engine,” in *Intelligent Computing Theories and Applications - 8th International Conference, ICIC 2012, Huangshan, China, July 25-29, 2012. Proceedings*, 2012, pp. 633–641. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-31576-3\\_80](http://dx.doi.org/10.1007/978-3-642-31576-3_80)
  144. A. Schätzle, M. Przyjaciół-Zablocki, A. Neu, and G. Lausen, “Sempala: Interactive SPARQL query processing on hadoop,” in *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, 2014, pp. 164–179. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-11964-9\\_11](http://dx.doi.org/10.1007/978-3-319-11964-9_11)
  145. G. Ladwig and A. Harth, “Cumulusrdf: Linked data management on nested key-value stores,” in *Proceedings of the 7th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2011) at the 10th International Semantic Web Conference (ISWC2011)*. -, Oktober 2011, Inproceedings.
  146. A. Corbellini, C. Mateos, A. Zunino, D. Godoy, and S. Schiaffino, “Persisting big-data: The nosql landscape,” *Information Systems*, vol. 63, pp. 1–23, 2017.
  147. D. Barbará, “Requirements for Clustering Data Streams,” *SIGKDD Explor. Newsl.*, pp. 23–27, 2002.
  148. J. a. Gama, *Knowledge Discovery from Data Streams*, ser. Chapman and Hall / Data Mining and Knowledge Discovery Series. CRC Press, 2010.
  149. M. Meir-Huber and M. Köhler, “Big data in austria.” Austrian Ministry for Transport, Innovation and Technology (BMVIT), Tech. Rep., 2014.
  150. M. V. Nural, H. Peng, and J. A. Miller, “Using meta-learning for model type selection in predictive big data analytics,” in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2027–2036.
  151. T. Cunha, C. Soares, and A. C. de Carvalho, “Meta-learning and recommender systems: A literature review and empirical study on the algorithm selection problem for collaborative filtering,” *Information Sciences*, vol. 423, pp. 128–144, 2018.
  152. G. Blair, N. Bencomo, and R. France, “Models@run.time,” *Computer*, vol. 42, no. 10, pp. 22–27, 2009.
  153. S. Schmid, I. Gerostathopoulos, C. Prehofer, and T. Bures, “Self-adaptation based on big data analytics: a model problem and tool,” in *Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2017 IEEE/ACM 12th International Symposium on*. IEEE, 2017, pp. 102–108.
  154. T. Hartmann, A. Moawad, F. Fouquet, G. Nain, J. Klein, and Y. L. Traon, “Stream my models: Reactive peer-to-peer distributed models@run.time,” in *Proceedings of the 18th International Conference on Model Driven Engineering Languages and Systems (MoDELS)*. ACM/IEEE, 2015.
  155. W. van der Aalst and E. Damiani, “Processes meet big data: Connecting data science with process science,” *IEEE Transactions on Services Computing*, vol. 8, no. 6, pp. 810–819, 2015.
  156. D. C. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, 2001.
  157. W. M. P. van der Aalst, “Process mining,” *Commun. ACM*, vol. 55, no. 8, pp. 76–83, 2012.
  158. W. M. P. van der Aalst, A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, R. P. J. C. Bose, P. van den Brand, R. Brandtjen, J. C. A. M. Buijs, A. Burattin, J. Carmona, M. Castellanos, J. Claes, J. Cook, N. Costantini, F. Curbera, E. Damiani, M. de Leoni, P. Delias, B. F. van Dongen, M. Dumas, S. Dustdar, D. Fahland, D. R. Ferreira, W. Gaaloul, F. van Geffen, S. Goel, C. W. Günther, A. Guzzo, P. Harmon, A. H. M. ter Hofstede, J. Hoogland, J. E. Ingvaldsen, K. Kato, R. Kuhn, A. Kumar, M. L. Rosa, F. M. Maggi, D. Malerba, R. S. Mans, A. Manuel, M. McCreesh, P. Mello, J. Mendling, M. Montali, H. R. M. Nezhad, M. zur Muehlen, J. Munoz-Gama, L. Pontieri, J. Ribeiro, A. Rozinat, H. S. Pérez, R. S. Pérez, M. Sepúlveda, J. Sinur, P. Soffer, M. Song, A. Sperduti, G. Stilo, C. Stoel, K. D. Swenson, M. Talamo, W. Tan, C. Turner, J. Vanthienen, G. Varvaressos, E. Verbeek, M. Verdonk, R. Vigo, J. Wang, B. Weber, M. Weidlich, T. Weijters, L. Wen, M. Westergaard, and M. T. Wynn, “Process mining manifesto,” in *Proceedings of the Business Process Management Workshops (BPM)*. Springer, 2011, pp. 169–194.
  159. M. Dumas, W. M. P. van der Aalst, and A. H. M. ter Hofstede, *Process-Aware Information Systems: Bridging People and Software Through Process Technology*. Wiley, 2005.
  160. B. F. van Dongen and W. M. P. van der Aalst, “A meta model for process mining data,” in *Proceedings of the International Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI) co-located with the 17th Conference on Advanced Information Systems Engineering (CAiSE)*, 2005.
  161. H. Al-Ali, E. Damiani, M. Al-Qutayri, M. Abu-Matar, and R. Mizouni, “Translating bpmn to business rules,” in *International Symposium on Data-Driven Process Discovery and Analysis*. Springer, 2016, pp. 22–36.
  162. G. Hripcsak and A. S. Rothschild, “Agreement, the f-measure, and reliability in information retrieval,” *Journal of the American Medical Informatics Association*, vol. 12, no. 3, pp. 296–298, 2005.
  163. O. Gilson, N. Silva, P. Grant, and M. Chen, “From web data to visualization via ontology mapping,” *Computer Graphics Forum*, vol. 27, no. 3, pp. 959–966. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2008.01230.x>
  164. K. Nazemi, D. Burkhardt, M. Breyer, C. Stab, and D. W. Fellner, “Semantic visualization cockpit: Adaptable composition of semantics-visualization techniques for knowledge-exploration,” in *International Association of Online Engineering (IAOE): International Conference Interactive Computer Aided Learning*, 2010, pp. 163–173.
  165. K. Nazemi, M. Breyer, J. Forster, D. Burkhardt, and A. Kuijper, “Interacting with semantics: A user-centered visualization adaptation based on semantics data,” in *Human Interface and the Management of Information. Interacting with Information*, M. J. Smith and

- G. Salvendy, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 239–248.
166. C. Melo, A. Mikheev, B. Le-Grand, and M.-A. Aulfare, “Cubix: A visual analytics tool for conceptual and semantic data,” in *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 894–897.
  167. C. Fluit, M. Sabou, and F. van Harmelen, *Ontology-Based Information Visualization: Toward Semantic Web Applications*. London: Springer London, 2006, pp. 45–58. [Online]. Available: [https://doi.org/10.1007/1-84628-290-X\\_3](https://doi.org/10.1007/1-84628-290-X_3)
  168. S. Krivov, R. Williams, and F. Villa, “Growl: A tool for visualization and editing of owl ontologies,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 54–57, 2007.
  169. D. Chu, D. A. Sheets, Y. Zhao, Y. Wu, J. Yang, M. Zheng, and G. Chen, “Visualizing hidden themes of taxi movement with semantic transformation,” in *Visualization Symposium (PacificVis), 2014 IEEE Pacific*. IEEE, 2014, pp. 137–144.
  170. T. Catarci, M. Scannapieco, M. Console, and C. Demetrescu, “My (fair) big data,” in *2017 IEEE International Conference on Big Data, Big-Data 2017, Boston, MA, USA, December 11-14, 2017*, 2017, pp. 2974–2979. [Online]. Available: <https://doi.org/10.1109/BigData.2017.8258267>
  171. Oracle, “The five most common big data integration mistakes to avoid,” 2015, white paper, accessed on 20 Jun, 2017. [Online]. Available: <http://er2016.cs.titech.ac.jp/program/panel.html>
  172. S. M. F. Ali and R. Wrembel, “From conceptual design to performance optimization of etl workflows: current state of research and open problems,” *The VLDB Journal*, 2017. [Online]. Available: <https://doi.org/10.1007/s00778-017-0477-2>
  173. C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, “Pig latin: a not-so-foreign language for data processing,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, 2008, pp. 1099–1110. [Online]. Available: <http://doi.acm.org/10.1145/1376616.1376726>
  174. S. Venkataraman, Z. Yang, D. Liu, E. Liang, H. Falaki, X. Meng, R. Xin, A. Ghodsi, M. J. Franklin, I. Stoica, and M. Zaharia, “Sparkr: Scaling R programs with spark,” in *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, 2016, pp. 1099–1104. [Online]. Available: <http://doi.acm.org/10.1145/2882903.2903740>
  175. B. Dinter, P. Gluchowski, and C. Schieder, “A stakeholder lens on metadata management in business intelligence and big data—results of an empirical investigation,” 2015.
  176. A. Yazici and R. George, *Fuzzy database modeling*, ser. Studies in Fuzziness and Soft Computing. Physica Verlag, 1999, vol. 26, iISBN 978-3-7908-1171-1.
  177. G. Shafer, *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976.
  178. B. Wanders, M. van Keulen, and P. van der Vet, “Uncertain groupings: Probabilistic combination of grouping data,” in *Proc. of DEXA*, ser. LNCS, vol. 9261. Springer, 2015, pp. 236–250, doi 10.1007/978-3-319-22849-5\_17.
  179. J. Huang, L. Antova, C. Koch, and D. Olteanu, “MayBMS: A probabilistic database management system,” in *Proc. of SIGMOD*. ACM, 2009, pp. 1071–1074, doi 10.1145/1559845.1559984.
  180. M. Thiele, U. Fischer, and W. Lehner, “Partition-based workload scheduling in living data warehouse environments,” *Information Systems*, vol. 34, no. 4-5, pp. 382–399, 2009.
  181. M. Angelini and G. Santucci, “Modeling incremental visualizations,” in *Proc. of the EuroVis Workshop on Visual Analytics (EuroVA13)*, 2013, pp. 13–17.
  182. H.-J. Schulz, M. Angelini, G. Santucci, and H. Schumann, “An enhanced visualization process model for incremental visualization,” *IEEE transactions on visualization and computer graphics*, vol. 22, no. 7, pp. 1830–1842, 2016.
  183. C. D. Stolper, A. Perer, and D. Gotz, “Progressive visual analytics: User-driven visual exploration of in-progress analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1653–1662, 2014.
  184. J.-D. Fekete and R. Primet, “Progressive analytics: A computation paradigm for exploratory data analysis,” *arXiv preprint arXiv:1607.05162*, 2016.
  185. B. Shneiderman and A. Aris, “Network visualization by semantic substrates,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 733–740, 2006.
  186. D. Wu, M. J. Greer, D. W. Rosen, and D. Schaefer, “Cloud manufacturing: Strategic vision and state-of-the-art,” *Journal of Manufacturing Systems*, vol. 32, no. 4, pp. 564–579, 2013.
  187. K. E. Martin, “Ethical issues in the big data industry,” *MIS Quarterly Executive*, vol. 14, p. 2, 2015.