Journal on Data Semantics
https://doi.org/10.1007/s13740-018-0086-2

ORIGINAL ARTICLE

Big Data Semantics

Paolo Ceravolo[1] · Antonia Azzini[2] · Marco Angelini[3] · Tiziana Catarci[3] · Philippe Cudré-Mauroux[4] · Ernesto Damiani[5] · Alexandra Mazak[6] · Maurice Van Keulen[7] · Mustafa Jarrar[8] · Giuseppe Santucci[3] · Kai-Uwe Sattler[9] · Monica Scannapieco[10] · Manuel Wimmer[6] · Robert Wrembel[11] · Fadi Zaraket[12]

Received: 17 April 2018 / Accepted: 20 April 2018

Abstract

Big Data technology has discarded traditional data modeling approaches as no longer applicable to distributed data processing. It is, however, largely recognized 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 members 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, 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 modeling [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

✉ Paolo Ceravolo
paolo.ceravolo@unimi.it

[1] Università Degli Studi di Milano, Milan, Italy

[2] Consortium for the Technology Transfer, C2T, Milan, Italy

[3] SAPIENZA University of Rome, Rome, Italy

[4] University of Fribourg, Fribourg, Switzerland

[5] EBTIC, Khalifa University, Abu Dhabi, UAE

[6] Vienna University of Technology, Wien, Austria

[7] University of Twente, Enschede, The Netherlands

[8] Birzeit University, Birzeit, Palestine

[9] TU Ilmenau, Ilmenau, Germany

[10] Directorate for Methodology and Statistical Design, Italian National Institute of Statistics (Istat), Rome, Italy

[11] Poznan University of Technology, Poznan, Poland

[12] American University of Beirut, Beirut, Lebanon

Published online: 23 May 2018

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 has 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 lack 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 on the FAIR principles [18] recommending procedures that generate Findable, Accessible, Interoperable, and Reusable 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, principles, 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 as follows. In Sects. 2 and 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 Sect. 4, we develop the central discussion of this paper. Then, in Sect. 5, we review the research perspectives that emerge from our discussion and, finally, in Sect. 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 standardization in Big Data management.[1]

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 impact 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

[1] The IEEE Big Data Governance and Metadata Management (BDGMM) group (http://standards.ieee.org/develop/indconn/BDGMM_index.html) aims at enabling data integration among heterogeneous datasets from diversified domain repositories and makes data discoverable, accessible, and usable through an actionable standard infrastructure.

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 are crucial steps. A preparation stage may also be required to increase the obfuscation level of the data, for preserving privacy, intellectual property, or strategical 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.

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 recognized 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 dataset. 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 are 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 organizes 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 recognized as a major factor hindering Big Data technologies [102].

According to the EU directive on data protection [103], personal data are not simply data stored in the entries of a repository, but any information that can be inferred by processing these entries in combination with 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.

Table 1 References organized by stages of the Big Data pipeline and Data Semantics dimensions

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 modeling	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			

Table 1 continued

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 modeling	Data interpretation and reporting	Data	Metadata	Data quality	Process representation
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	x			
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		

Table 1 continued

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 modeling	Data interpretation and reporting	Data	Metadata	Data quality	Process representation
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

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 is also provided. In Fig. 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 languages 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 subgraphs 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 multilingual (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 MULText.

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 identifies mapping 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 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 with 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 semiautomatically allow to annotate data with quality scores (see [46]). In Fig. 1, the **quality metadata extraction and annotation task** include 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, 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 gray 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 Fig. 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 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 are processed in parallel (typically, by a MapReduce-like parallelization 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 toward an obser-

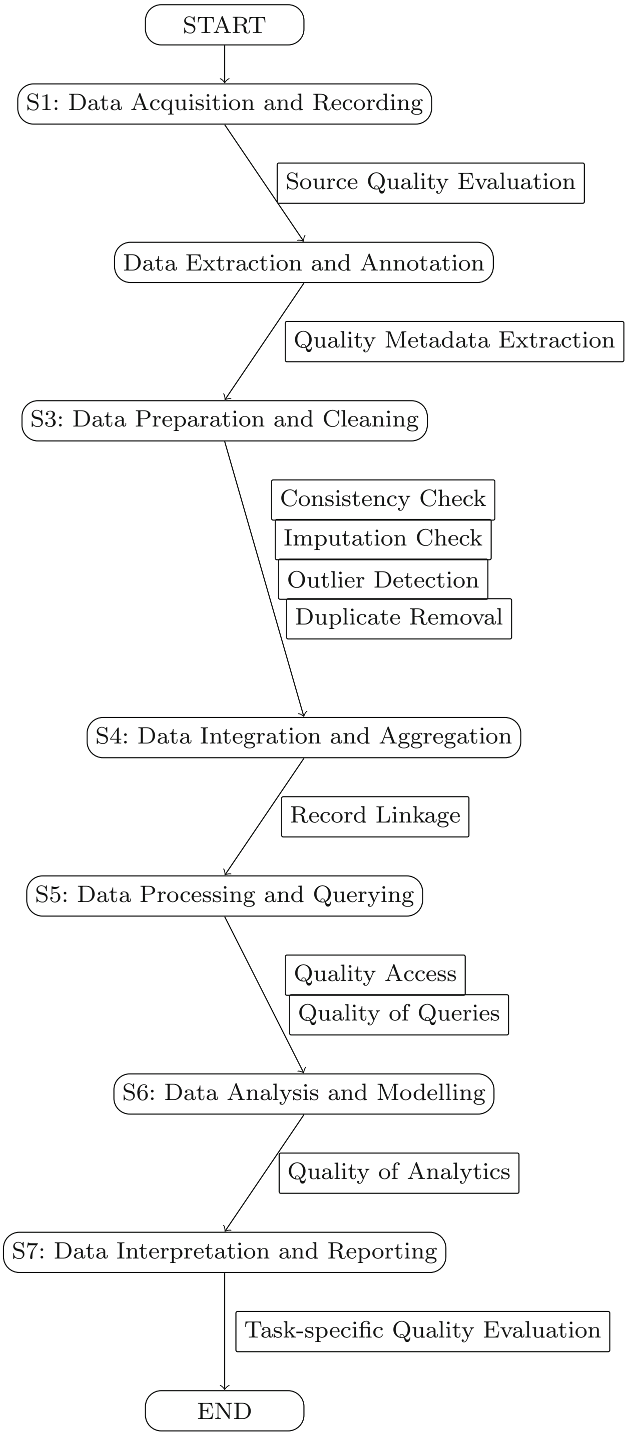
START
S1: Data Acquisition and Recording
Source Quality Evaluation
Data Extraction and Annotation
Quality Metadata Extraction
S3: Data Preparation and Cleaning
Consistency Check
Imputation Check
Outlier Detection
Duplicate Removal
S4: Data Integration and Aggregation
Record Linkage
S5: Data Processing and Querying
Quality Access
Quality of Queries
S6: Data Analysis and Modelling
Quality of Analytics
S7: Data Interpretation and Reporting
Task-specific Quality Evaluation
END

Fig. 1 Main stages of a Big Data pipeline and related data quality tasks

vation 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 incremental 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.[2] 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, and SAP Vora. Such technologies offer a means for querying jointly an enterprise data warehouse and a data lake.

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

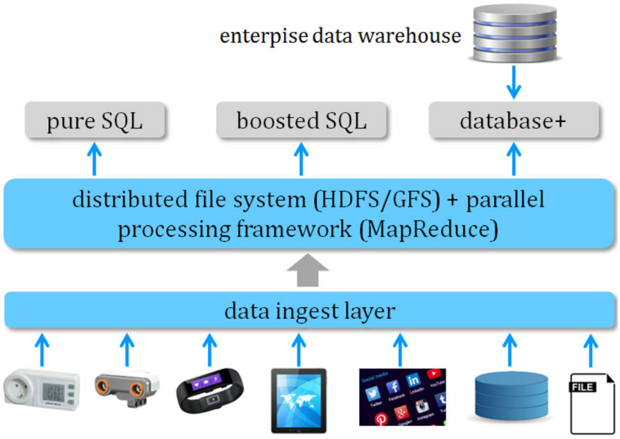
enterpise data warehouse
pure SQL
boosted SQL
database+
distributed file system (HDFS/GFS) + parallel processing framework (MapReduce)
data ingest layer
FILE

Fig. 2 An overall architecture of a data lake

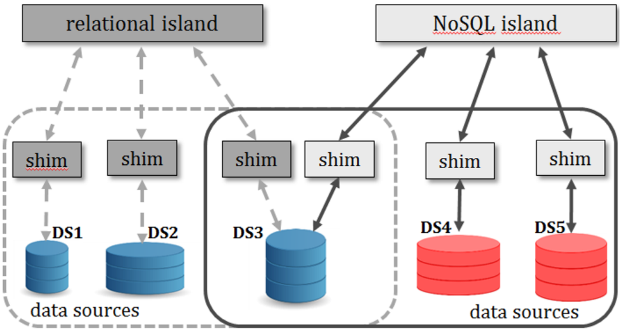

Fig. 3 An example architecture of the polystore with two islands of information

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

Recently, another Big Data integration architecture was proposed in [36], which is called a *polystore*. The main idea behind it is to organize datasets into the 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 Fig. 3. It is composed of two information islands: a relational one and a NoSQL one. The first island is composed of relational data sources *DS1*, *DS2*, and *DS3*, each of which has its dedicated shim that exposes a relational interface to the island. The second island is composed of NoSQL data sources *DS4* and *DS5*. Their shims expose a NoSQL interface to the NoSQL island. Notice that *DS3* is shared by both islands through 2 shims, i.e., it can be queried from both islands.

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[3] or DBPedia[4]) or private (e.g., Google's Knowledge Graph.[5]) 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 datasets, 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].

[3] https://www.wikidata.org.

[4] http://wiki.dbpedia.org/.

[5] https://www.google.com/intl/en-419/insidesearch/features/search/knowledge.html.

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 Fig. 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, 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 support a broad range of data access patterns for different kinds of queries and analytics. Several approaches have been developed toward 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 and CSV, 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.

The core of Spark[6] is 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.[7] 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 and a declarative API that integrates with procedural code. It includes an extensible

[6] https://spark.apache.org/.

[7] http://mesos.apache.org/.

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 Fig. 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 Modeling

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 are 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 combined [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 does not involve the definition of an end-to-end process, 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, and to streamline processes [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 modeling phase does also have an important quality counterpart, **quality analysis**, c.f. Fig. 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 pose 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 aforementioned 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, Voigt et al. [91] extract, 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 [70,92]. 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 Fig. 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, [8] AirBnB Airflow, [9] LinkedIn Azkaban, [10], and Spring Cloud Data Flow. [11] These frameworks 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 reusability 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.

[8] http://oozie.apache.org.

[9] https://airflow.apache.org.

[10] https://azkaban.github.io.

[11] https://cloud.spring.io/spring-cloud-dataflow/.

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 dataflow abstraction for modeling complex pipelines is proposed in [100]. The dataflows 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 propose 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.[12]

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 for producing clean and well 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 dataset 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 online 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 combine 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) toward declarative and optimizable query languages such as SQL dialects.
- Developing a **metadata standard** and architecture. The latter should support: automatic or semiautomatic 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 [102,175]. Metadata management opens challenges that affect almost all aspects of Big Data [26]. For example, the data processing engine has to identify the datasets that can be used for starting the ingestion procedure, as there may exist multiple datasets storing semantically the same data. The problem is then to figure out which datasets 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,

[12] http://www.toreador-project.eu.

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 online 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 are 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) that 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, 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 semiautomated 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 multiparty 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. Zikopoulos P, Eaton C et al (2011) Understanding big data: analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media, New York
2. Ward JS, Barker A (2013) Undefined by data: a survey of big data definitions. arXiv preprint arXiv:1309.5821
3. Beyer MA, Laney D (2012) The importance of big data: a definition. Gartner, Stamford, pp 2014–2018
4. Laney D (2001) 3d data management: controlling data volume, velocity and variety. META Gr Res Note 6:70
5. Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Khan SU (2015) The rise of "big data" on cloud computing: review and open research issues. Inf Syst 47:98–115
6. Wamba SF, Akter S, Edwards A, Chopin G, Gnanzou D (2015) How big data can make big impact: findings from a systematic review and a longitudinal case study. Int J Prod Econ 165:234–246 [Online]. http://www.sciencedirect.com/science/article/pii/S0925527314004253. Accessed 20 Feb 2018
7. Madden S (2012) From databases to big data. IEEE Internet Comput 16(3):4–6
8. Amazon A (2016) Amazon 2016 [Online]. https://aws.amazon.com. 2016-01-06
9. Hadoop A (2009) Hadoop [Online]. http://hadoop.apache.org. 2009-03-06
10. Chen H, Chiang RH, Storey VC (2012) Business intelligence and analytics: from big data to big impact. MIS Q 36(4):1165–1188
11. Wu X, Zhu X, Wu G-Q, Ding W (2014) Data mining with big data. IEEE Trans Knowl Data Eng 26(1):97–107
12. Hilbert M (2016) Big data for development: a review of promises and challenges. Dev Policy Rev 34(1):135–174
13. Assunç ao MD, Calheiros RN, Bianchi S, Netto MA, Buyya R, (2015) Big data computing and clouds: trends and future directions. J Parallel Distrib Comput 79:3–15
14. Markl V (2014) Breaking the chains: on declarative data analysis and data independence in the big data era. Proc VLDB Endow 7(13):1730–1733
15. Damiani E, Oliboni B, Quintarelli E, Tanca L (2003) Modeling semistructured data by using graph-based constraints. OTM confederated international conferences "On the move to meaningful internet systems". Springer, Berlin, pp 20–21
16. Poole J, Chang D, Tolbert D, Mellor D (2003) Common warehouse metamodel. Developer's guide, Wiley, Hoboken
17. Ardagna C, Asal R, Damiani E, Vu Q (2015) From security to assurance in the cloud: a survey. ACM Comput Surv: CSUR 48(1):2:1–2:50
18. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J-W, da Silva Santos LB, Bourne PE et al (2016) The fair guiding principles for scientific data management and stewardship. Sci Data 3:160018
19. Aberer K, Catarci T, Cudré-Mauroux P, Dillon T, Grimm S, Hacid M-S, Illarramendi A, Jarrar M, Kashyap V, Mecella M et al (2004) Emergent semantics systems. Semantics of a networked world. Semantics for grid databases. Springer, Berlin, pp 14–43
20. Cudré-Mauroux P, Aberer K, Abdelmoty AI, Catarci T, Damiani E, Illaramendi A, Jarrar M, Meersman R, Neuhold EJ, Parent C et al (2006) Viewpoints on emergent semantics. In: Spaccapietra S, Aberer K, Cudré-Mauroux P (eds) Journal on data semantics VI. Springer, Berlin, pp 1–27
21. Ardagna CA, Ceravolo P, Damiani E (2016) Big data analytics as-a-service: Issues and challenges. In: IEEE International conference on Big Data (Big Data). IEEE, pp 3638–3644
22. Chen M, Mao S, Liu Y (2014) Big data: a survey. Mob Netw Appl 19(2):171–209
23. Azzini A, Ceravolo P (2013) Consistent process mining over big data triple stores. In: IEEE international congress on Big Data (BigData Congress). IEEE, pp 54–61
24. Woods WA (1975) What's in a link: foundations for semantic networks. In: Representation and understanding. Elsevier, pp 35–82
25. Franklin MJ, Halevy AY, Maier D (2005) From databases to dataspaces: a new abstraction for information management. SIGMOD Rec 34(4):27–33 [Online]. https://doi.org/10.1145/1107499.1107502
26. Smith K, Seligman L, Rosenthal A, Kurcz C, Greer M, Macheret C, Sexton M, Eckstein A (2014) Big metadata: the need for principled metadata management in big data ecosystems. In: Proceedings of workshop on data analytics in the Cloud, series DanaC'14. ACM, New York, pp 13:1–13:4 [Online]. https://doi.org/10.1145/2627770.2627776
27. Waller MA, Fawcett SE (2013) Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management. J Bus Logist 34(2):77–84
28. Borkar V, Carey MJ, Li C (2012) Inside big data management: ogres, onions, or parfaits? In: Proceedings of the 15th international conference on extending database technology. ACM, pp 3–14
29. White T (2012) Hadoop: the definitive guide. O'Reilly Media Inc, Sebastopol
30. Jagadish H (2015) Big data and science: myths and reality. Big Data Res 2(2):49–52
31. Pääkkönen P, Pakkala D (2015) Reference architecture and classification of technologies, products and services for big data systems. Big Data Res 2(4):166–186
32. Ardagna C, Bellandi V, Bezzi M, Ceravolo P, Damiani E, Hebert C (June 2017) A model-driven methodology for big data analytics-as-a-service. In: Proceedings of BigData Congress, Honolulu. HI, USA
33. Labrinidis A, Jagadish HV (2012) Challenges and opportunities with big data. Proc VLDB Endow 5(12):2032–2033. https://doi.org/10.14778/2367502.2367572

34. Ardagna CA, Bellandi V, Bezzi M, Ceravolo P, Damiani E, Hebert C (2018) Model-based big data analytics-as-a-service: take big data to the next level. IEEE Trans Serv Comput PP(99):1–1
35. Liao C, Squicciarini A (2015) Towards provenance-based anomaly detection in mapreduce. In: 15th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid), vol 2015. IEEE, pp 647–656
36. Duggan J, Elmore AJ, Stonebraker M, Balazinska M, Howe B, Kepner J, Madden S, Maier D, Mattson T, Zdonik S (2015) The BigDAWG polystore system. SIGMOD Rec 44(2):11–16
37. Sowmya R, Suneetha K (2017) Data mining with big data. In: 11th international conference on intelligent systems and control (ISCO). IEEE, pp 246–250
38. Zhou W, Mapara S, Ren Y, Li Y, Haeberlen A, Ives Z, Loo BT, Sherr M (2012) Distributed time-aware provenance. In: Proceedings of the VLDB endowment, vol 6, no 2. VLDB Endowment, pp 49–60
39. Akoush S, Sohan R, Hopper A (2013) Hadoopprov: towards provenance as a first class citizen in mapreduce. In: TaPP
40. Glavic B (2014) Big data provenance: challenges and implications for benchmarking. In: Rabl T, Poess M, Baru C, Jacobsen H-A (eds) Specifying big data benchmarks. Springer, Berlin, Heidelberg, pp 72–80
41. Berti-Equille L, Ba ML (2016) Veracity of big data: challenges of cross-modal truth discovery. J. Data Inf Qual 7(3):12:1–12:3
42. Kläs M, Putz W, Lutz T (2016) Quality evaluation for big data: a scalable assessment approach and first evaluation results. In: 2016 joint conference of the international workshop on software measurement and the international conference on software process and product measurement (IWSM-MENSURA). IEEE, pp 115–124
43. Daiber J, Jakob M, Hokamp C, Mendes PN (2013) Improving efficiency and accuracy in multilingual entity extraction. In: Proceedings of the 9th international conference on semantic systems. ACM, pp 121–124
44. Shin J, Wu S, Wang F, De Sa C, Zhang C, Ré C (July 2015) Incremental knowledge base construction using DeepDive. Proc VLDB Endow 8(11), 1310–1321. ISSN 2150-8097. https://doi.org/10.14778/2809974.2809991
45. Chiticariu L, Krishnamurthy R, Li Y, Raghavan S, Reiss FR, Vaithyanathan S (2010) Systemt: an algebraic approach to declarative information extraction. In: Proceedings of the association for computational linguistics, pp 128–137
46. Fuhring P, Naumann F (2007) Emergent data quality annotation and visualization [Online]. https://hpi.de/fileadmin/user_upload/fachgebiete/naumann/publications/2007/Emergent_Data_Quality_Annotation_and_Visualization.pdf. Accessed 20 Feb 2018
47. Bondiombouy C, Kolev B, Levchenko O, Valduriez P (2016) Multistore big data integration with CloudMdsQL. In: Hameurlain A, Küng J, Wagner R, Chen Q (eds) Transactions on large-scale data-and knowledge-centered systems XXVIII: special issue on database-and expert-systems applications. Springer, Berlin, Heidelberg, pp 48–74. https://doi.org/10.1007/978-3-662-53455-7_3
48. Bergamaschi S, Beneventano D, Mandreoli F, Martoglia R, Guerra F, Orsini M, Po L, Vincini M, Simonini G, Zhu S , Gagliardelli L, Magnotta L (2018) From data integration to big data integration. In: Flesca S, Greco S, Masciari E, Saccà D (eds) A comprehensive guide through the Italian database research over the last 25 years. Springer, Cham, pp 43–59
49. Ramakrishnan R, Sridharan B, Douceur JR, Kasturi P, Krishnamachari-Sampath B, Krishnamoorthy K, Li P, Manu M, Michaylov S, Ramos R et al (2017) 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, pp 51–63
50. Masseroli M, Kaitoua A, Pinoli P, Ceri S (2016) Modeling and interoperability of heterogeneous genomic big data for integrative processing and querying. Methods 111:3–11
51. Scannapieco M, Virgillito A, Zardetto D (2013) Placing big data in official statistics: a big challenge? In: Proceedings of NTTS (new techniques and technologies for statistics), March 5–7, Brussels
52. Gualtieri M, Hopkins B (2014) SQL-For-Hadoop: 14 capable solutions reviewed. Forrester
53. Liu H, Kumar TA, Thomas JP (2015) Cleaning framework for big data-object identification and linkage. In: IEEE international congress on Big Data (BigData Congress). IEEE, pp 215–221
54. Gulzar MA, Interlandi M, Han X, Li M, Condie T, Kim M (2017) Automated debugging in data-intensive scalable computing. In: Proceedings of the 2017 symposium on cloud computing, series SoCC ’17. ACM, New York, pp 520–534 [Online]. https://doi.org/10.1145/3127479.3131624
55. de Wit T (2017) Using AIS to make maritime statistics. In: Proceedings of NTTS (New techniques and technologies for statistics), March 14–16, Brussels
56. Zardetto D, Scannapieco M, Catarci T (2010) Effective automated object matching. In: Proceedings of the 26th international conference on data engineering, ICDE 2010, March 1-6, Long Beach, California, USA, pp 757–768
57. Xin RS, Gonzalez JE, Franklin MJ, Stoica I (2013) Graphx: a resilient distributed graph system on spark. In: First international workshop on graph data management experiences and systems, GRADES 2013, co-loated with SIGMOD/PODS, New York, NY, USA, June 24, p 2 [Online]. http://event.cwi.nl/grades2013/02-xin.pdf. Accessed 20 Feb 2018
58. Junghanns M, Petermann A, Gómez K, Rahm E (2015) GRADOOP: scalable graph data management and analytics with hadoop. CoRR [Online]. arxiv:1506.00548
59. Yu J, Wu J, Sarwat M (2015) 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, pp 70:1–70:4 [Online]. https://doi.org/10.1145/2820783.2820860
60. You S, Zhang J, Gruenwald L (2015) Large-scale spatial join query processing in cloud. In: 31st IEEE international conference on data engineering workshops, ICDE workshops 2015, Seoul, South Korea, April 13–17, pp 34–41. [Online]. https://doi.org/10.1109/ICDEW.2015.7129541
61. Saleh O, Hagedorn S, Sattler K (2015) Complex event processing on linked stream data. Datenbank Spektrum 15(2):119–129
62. Kornacker M, Behm A, Bittorf V, Bobrovytsky T, Ching C, Choi A, Erickson J, Grund M, Hecht D, Jacobs M, Joshi I, Kuff L, Kumar D, Leblang A, Li N, Pandis I, Robinson H, Rorke D, Rus S, Russell J, Tsirogiannis D, Wanderman-Milne S, Yoder M (2015) 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, Online proceedings, 2015 [Online]. http://www.cidrdb.org/cidr2015/Papers/CIDR15_Paper28.pdf
63. Costea A, Ionescu A, Raducanu B, Switakowski M, Bârca C, Sompolski J, Luszczak A, Szafranski M, de Nijs G, Boncz PA (2016) 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, pp 1105–1117 [Online]. https://doi.org/10.1145/2882903.2903742
64. Schätzle A, Przyjaciel-Zablocki M, Skilevic S, Lausen G (2016) S2RDF: RDF querying with SPARQL on spark. PVLDB

9(10):804–815 [Online]. http://www.vldb.org/pvldb/vol9/p804-schaetzle.pdf

65. Cudré-Mauroux P, Enchev I, Fundatureanu S, Groth PT, Haque A, Harth A, Keppmann FL, Miranker DP, Sequeda J, Wylot M (2013) Nosql databases for RDF: an empirical evaluation. In: The semantic Web—ISWC 2013—12th international semantic web conference, Sydney, NSW, Australia, October 21–25, Proceedings, Part II, 2013, pp 310–325 [Online]. https://doi.org/10.1007/978-3-642-41338-4_20
66. Appice A, Ceci M, Malerba D (2018) Relational data mining in the era of big data. In: Flesca S, Greco S, Masciari E, Saccà D (eds) A comprehensive guide through the Italian database research over the last 25 years. Springer, cham, pp 323–339. https://doi.org/10.1007/978-3-319-61893-7_19
67. Khare S, An K, Gokhale AS, Tambe S, Meena A (2015) Reactive stream processing for data-centric publish/subscribe. In: Proceedings of the 9th international conference on distributed event-based systems (DEBS). ACM, pp 234–245
68. Poggi F, Rossi D, Ciancarini P, Bompani L (2016) Semantic runtime models for self-adaptive systems: a case study. In: 2016 IEEE 25th international conference on enabling technologies: infrastructure for collaborative enterprises (WETICE). IEEE, pp 50–55
69. Um J-H, Lee S, Kim T-H, Jeong C-H, Song S-K, Jung H (2016) Semantic complex event processing model for reasoning research activities. Neurocomputing 209:39–45
70. Giese M, Soylu A, Vega-Gorgojo G, Waaler A, Haase P, Jiménez-Ruiz E, Lanti D, Rezk M, Xiao G, Özçep Ö et al (2015) Optique: zooming in on big data. Computer 48(3):60–67
71. Unece big data quality framework [Online]. http://www1.unece.org/stat/platform/display/bigdata/2014+Project. Accessed 20 Feb 2018
72. Severin J, Lizio M, Harshbarger J, Kawaji H, Daub CO, Hayashizaki Y, Bertin N, Forrest AR, Consortium F et al (2014) Interactive visualization and analysis of large-scale sequencing datasets using zenbu. Nat Biotechnol 32(3):217–219
73. Mezghani E, Exposito E, Drira K, Da Silveira M, Pruski C (2015) A semantic big data platform for integrating heterogeneous wearable data in healthcare. J Med Syst 39(12):185
74. Ginsberg J, Mohebbi M, Patel R, Brammer L, Smolinski M, Brilliant L (2009) Detecting influenza epidemics using search engine query data. Nature 457(7232):1012–1014
75. Sculley D, Holt G, Golovin D, Davydov E, Phillips T, Ebner D, Chaudhary V, Young M, Crespo J-F, Dennison D (2015) Hidden technical debt in machine learning systems. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems 28, Curran Associates, Inc., pp 2503–2511. http://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf
76. Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, Burrows M, Chandra T, Fikes A, Gruber RE (2008) Bigtable: a distributed storage system for structured data. ACM Trans Comput Syst 26(2):4:1–4:26. https://doi.org/10.1145/1365815.1365816
77. Suriarachchi I, Plale B (2016) Provenance as essential infrastructure for data lakes. In: Proceedings of international workshop on provenance and annotation of data and processes. LNCS 9672
78. Terrizzano I, Schwarz P, Roth M, Colino JE (2015) Data wrangling: the challenging journey from the wild to the lake. In: Proceedings of conference on innovative data systems research (CIDR)
79. Teradata (2014) Putting the data lake to work: a guide to best practices. http://www.teradata.com/Resources/Best-Practice-Guides/Putting-the-Data-Lake-to-Work-A-Guide-to-Bes. Accessed on 20 June 2017 [Online]
80. Batini C, Scannapieco M (2016) Data and information quality—dimensions. Principles and techniques, series. In: Data-centric systems and applications. Springer
81. Agrawal D, Bernstein P, Bertino E, Davidson S, Dayal U, Franklin M, Gehrke J, Haas L, Halevy A, Han J et al (2011) Challenges and opportunities with big data. Purdue University, Cyber Center Technical Reports
82. Liu M, Wang Q (2016) Rogas: a declarative framework for network analytics. Proceedings of international conference on very large data bases (VLDB) 9(13):1561–1564
83. Hasan O, Habegger B, Brunie L, Bennani N, Damiani E (2013) A discussion of privacy challenges in user profiling with big data techniques: the EEXCESS use case. In: IEEE international congress on Big Data (BigData Congress). IEEE, pp 25–30
84. Doan A, Ardalan A, Ballard JR, Das S, Govind Y, Konda P, Li H, Paulson E, Zhang H et al (2017) Toward a system building agenda for data integration. arXiv preprint arXiv:1710.00027
85. Flood M, Grant J, Luo H, Raschid L, Soboroff I, Yoo K (2016) 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, p 1
86. Haryadi AF, Hulstijn J, Wahyudi A, Van Der Voort H, Janssen M (2016) Antecedents of big data quality: an empirical examination in financial service organizations. In: IEEE international conference on Big Data (Big Data). IEEE, pp 116–121
87. Benedetti F, Beneventano D, Bergamaschi S (2016) Context semantic analysis: a knowledge-based technique for computing inter-document similarity. Springer International Publishing, Berlin, pp 164–178
88. Ford E, Carroll JA, Smith HE, Scott D, Cassell JA (2016) Extracting information from the text of electronic medical records to improve case detection: a systematic review. J Am Med Inform Assoc 23(5):1007–1015. https://doi.org/10.1093/jamia/ocv180
89. Haas D, Krishnan S, Wang J, Franklin MJ, Wu E (2015) Wisteria: nurturing scalable data cleaning infrastructure. Proc VLDB Endow 8(12):2004–2007. https://doi.org/10.14778/2824032.2824122
90. Cabot J, Toman D, Parsons J, Pastor O, Wrembel R (2016) Big data and conceptual models: are they mutually compatible? In: International conference on conceptual modeling (ER), panel discussion [Online]. http://er2016.cs.titech.ac.jp/program/panel.html. Accessed 20 Feb 2018
91. Voigt M, Pietschmann S, Grammel L, Meißner K (2012) Context-aware recommendation of visualization components. In: Proceedings of the 4th international conference on information, process, and knowledge management. Citeseer, pp 101–109
92. Soylu A, Giese M, Jimenez-Ruiz E, Kharlamov E, Zheleznyakov D, Horrocks I (2013) OptiqueVQS: towards an ontology-based visual query system for big data. In: Proceedings of the fifth international conference on management of emergent digital ecosystems, series, MEDES ’13. ACM, New York, pp 119–126 [Online]. https://doi.org/10.1145/2536146.2536149
93. McKenzie G, Janowicz K, Gao S, Yang J-A, Hu Y (2015) POI pulse: a multi-granular, semantic signature-based information observatory for the interactive visualization of big geosocial data. Cartographica Int J Geogr Inf Geovis 50(2):71–85
94. Habib MB, Van Keulen (2016) TwitterNEED: a hybrid approach for named entity extraction and disambiguation for tweet. Nat Lang Eng 22(3):423–456. https://doi.org/10.1017/S1351324915000194
95. Magnani M, Montesi D (2010) A survey on uncertainty management in data integration. JDIQ 2(1):5:1–5:33. https://doi.org/10.1145/1805286.1805291
96. van Keulen M (2012) Managing uncertainty: the road towards better data interoperability. Inf Technol: IT 54(3):138–146. https://doi.org/10.1524/itit.2012.0674

97. Andrews P, Kalro A, Mehanna H, Sidorov A (2016) Productionizing machine learning pipelines at scale. In: Machine learning systems workshop at ICML
98. Sparks ER, Venkataraman S, Kaftan T, Franklin MJ, Recht B (2017) Keystoneml: optimizing pipelines for large-scale advanced analytics. In: 2017 IEEE 33rd international conference on data engineering (ICDE), pp 535–546
99. Meng X, Bradley J, Yavuz B, Sparks E, Venkataraman S, Liu D, Freeman J, Tsai D, Amde M, Owen S et al (2016) Mllib: machine learning in apache spark. J Mach Learn Res 17(1):1235–1241
100. Böse J-H, Flunkert V, Gasthaus J, Januschowski T, Lange D, Salinas D, Schelter S, Seeger M, Wang Y (2017) Probabilistic demand forecasting at scale. Proc VLDB Endow 10(12):1694–1705
101. Baylor D, Breck E, Cheng H-T, Fiedel N, Foo CY, Haque Z, Haykal S, Ispir M, Jain V, Koc L et al (2017) 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, pp 1387–1395
102. Ardagna C, Ceravolo P, Cota GL, Kiani MM, Damiani E (2017) What are my users looking for when preparing a big data campaign. In: IEEE international congress on Big Data (BigData Congress). IEEE, pp 201–208
103. Palmér C (2017) Modelling eu directive 2016/680 using enterprise architecture
104. Atzmueller M, Kluegl P, Puppe F (2008) Rule-based information extraction for structured data acquisition using textmarker. In: Proceedings of LWA, pp 1–7
105. Settles B (2011) Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In: Proceedings of EMNLP.ACL, pp 1467–1478
106. Müller C, Strube M (2006) Multi-level annotation of linguistic data with MMAX2. Corpus Technol Lang Pedag New Resour New Tools New Methods 3:197–214
107. Stenetorp P, Pyysalo S, Topić G, Ohta T, Ananiadou S, Tsujii J (2012) Brat: a web-based tool for NLP-assisted text annotation. In: Proceedings of the demonstrations at the 13th conference of the European chapter of the association for computational linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 102–107
108. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z (2007) DBpedia: a nucleus for a web of open data. In: Aberer K, Choi K-S, Noy N, Allemang D, Lee K, Nixon L, Golbeck J, Mika P, Maynard D, Mizoguchi R, Schreiber G, Cudré-Mauroux P (eds) The semantic web. Springer, Berlin, Heidelberg, pp 722–735
109. Bizer C, Heath T, Berners-Lee T (2009) Linked data–the story so far. Int J Semant Web Inf Syst: IJSWIS 5(3):1–22
110. Benikova D, Biemann C (2016) Semreldata ? Multilingual contextual annotation of semantic relations between nominals: dataset and guidelines. In: LREC
111. Lu A, Wang W, Bansal M, Gimpel K, Livescu K (2015) Deep multilingual correlation for improved word embeddings. In: NAACL-HLT
112. Pecina P, Toral A, Way A, Papavassiliou V, Prokopidis P, Giagkou M (2011) Towards using web-crawled data for domain adaptation in statistical machine translation. In: The 15th conference of the European association for machine translation (EAMT)
113. Yasseri T, Spoerri A, Graham M, Kertész J (2014) The most controversial topics in Wikipedia: a multilingual and geographical analysis. In: Fichman P, Hara N (eds) Global Wikipedia: international and cross-cultural issues in online collaboration. Rowman & Littlefield Publishers Inc, Lanham, pp 25–48
114. Micher JC (2012) Improving domain-specific machine translation by constraining the language model. Army Research Laboratory, Technical Report of ARL-TN-0492
115. D'Haen J, den Poel DV, Thorleuchter D, Benoit D (2016) Integrating expert knowledge and multilingual web crawling data in a lead qualification system. Decis Support Syst 82:69–78
116. Helou MA, Palmonari M, Jarrar M (2016) Effectiveness of automatic translations for cross-lingual ontology mapping. J Artif Int Res 55(1):165–208
117. Furno D, Loia V, Veniero M, Anisetti M, Bellandi V, Ceravolo P, Damiani E (2011) Towards an agent-based architecture for managing uncertainty in situation awareness. In: 2011 IEEE symposium on intelligent agent (IA). IEEE, pp 1–6
118. Dalvi N, Ré C, Suciu D (2009) Probabilistic databases: diamonds in the dirt. Commun ACM 52(7):86–94. https://doi.org/10.1145/1538788.1538810
119. Ceravolo P, Damiani E, Fugazza C (2007) 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], pp 131–132
120. Panse F, van Keulen M, Ritter N (2013) Indeterministic handling of uncertain decisions in deduplication. JDIQ 4(2):91–925. https://doi.org/10.1145/2435221.2435225
121. Abedjan Z, Golab L, Naumann F (2015) Profiling relational data: a survey. VLDB J 24(4):557–581. https://doi.org/10.1007/s00778-015-0389-y
122. Papenbrock T, Ehrlich J, Marten J, Neubert T, Rudolph J-P, Schönberg M, Zwiener J, Naumann F (2015) Functional dependency discovery: an experimental evaluation of seven algorithms. Proc VLDB Endow 8(10):1082–1093
123. Chen CLP, Zhang C (2014) Data-intensive applications, challenges, techniques and technologies: a survey on big data. Inf Sci 275:314–347
124. Naumann F (2014) Data profiling revisited. SIGMOD Rec 42(4):40–49
125. Ahmadov A, Thiele M, Eberius J, Lehner W, Wrembel R (2015) Towards a hybrid imputation approach using web tables. In: IEEE/ACM international symposium on big data computing (BDC), pp 21–30
126. Ahmadov A, Thiele M, Lehner W, Wrembel R (2017) Context similarity for retrieval-based imputation. In: International symposium on foundations and applications of big data analytics (FAB) **(to appear)**
127. Li Z, Sharaf MA, Sitbon L, Sadiq S, Indulska M, Zhou X (2014) A web-based approach to data imputation. World Wide Web 17(5):873–897
128. Miao X, Gao Y, Guo S, Liu W (2018) Incomplete data management: a survey. Front Comput Sci 12(1):4–25. https://doi.org/10.1007/s11704-016-6195-x
129. Wiederhold G (1992) Mediators in the architecture of future information systems. IEEE Comput 25(3):38–49
130. Tonon A, Demartini G, Cudré-Mauroux P (2012) 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, pp 125–134 [Online]. https://doi.org/10.1145/2348283.2348304
131. Catasta M, Tonon A, Demartini G, Ranvier J, Aberer K, Cudré-Mauroux P (2014) B-hist: entity-centric search over personal web browsing history. J Web Semant 27:19–25 [Online]. https://doi.org/10.1016/j.websem.2014.07.003
132. Flood M, Jagadish HV, Raschid L (2016) Big data challenges and opportunities in financial stability monitoring. Financ Stab Rev 20:129–142
133. Ni LM, Tan H, Xiao J (2016) Rethinking big data in a networked world. Front Comput Sci 10(6):965–967
134. Kolb L, Thor A, Rahm E (2012) Dedoop: efficient deduplication with hadoop. PVLDB 5(12):1878–1881

135. Ghemawat S, Gobioff H, Leung S (2003) 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, pp 29–43 [Online]. https://doi.org/10.1145/945445.945450
136. Dittrich J, Quiané-Ruiz J, Richter S, Schuh S, Jindal A, Schad J (2012) Only aggressive elephants are fast elephants. PVLDB 5(11):1591–1602 [Online]. http://vldb.org/pvldb/vol5/p1591_jensdittrich_vldb2012.pdf
137. Carbone P, Katsifodimos A, Ewen S, Markl V, Haridi S, Tzoumas K (2015) Apache flink™: stream and batch processing in a single engine. IEEE Data Eng Bull 38(4):28–38 [Online]. http://sites.computer.org/debull/A15dec/p28.pdf
138. Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauly M, Franklin MJ, Shenker S, Stoica I (2012) 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, pp 15–28 [Online]. https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/zaharia. Accessed 20 Feb 2018
139. Armbrust M, Xin RS, Lian C, Huai Y, Liu D, Bradley JK, Meng X, Kaftan T, Franklin MJ, Ghodsi A, Zaharia M (2015) 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, pp 1383–1394 [Online]. https://doi.org/10.1145/2723372.2742797
140. Hagedorn S, Götze P, Sattler K (2017) The STARK framework for spatio-temporal data analytics on spark. In: Datenbanksysteme für Business, Technologie und Web (BTW, 17. Fachtagung des GI-Fachbereichs, Datenbanken und Informationssysteme" (DBIS), 6.-10. März 2017. Stuttgart, Germany, Proceedings, pp 123–142
141. Meng X, Bradley JK, Yavuz B, Sparks ER, Venkataraman S, Liu D, Freeman J, Tsai D B, Amde M, Owen S, Xin D, Xin R, Franklin MJ, Zadeh R, Zaharia M, Talwalkar A (2016) Mllib: machine learning in apache spark. J Mach Learn Res 17:34:1–34:7 [Online]. http://jmlr.org/papers/v17/15-237.html
142. Abouzeid A, Bajda-Pawlikowski K, Abadi DJ, Rasin A, Silberschatz A (2009) Hadoopdb: an architectural hybrid of mapreduce and DBMS technologies for analytical workloads. PVLDB 2(1):922–933 [Online]. http://www.vldb.org/pvldb/2/vldb09-861.pdf
143. Du J, Wang H, Ni Y, Yu Y (2012) Hadooprdf: a scalable semantic data analytical engine. In: Intelligent computing theories and applications—8th international conference, ICIC 2012, Huangshan, China, July 25–29. Proceedings, pp 633–641 [Online]. https://doi.org/10.1007/978-3-642-31576-3_80
144. Schätzle A, Przyjaciel-Zablocki M, Neu A, Lausen G (2014) 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. Proceedings, Part I, pp 164–179 [Online]. https://doi.org/10.1007/978-3-319-11964-9_11
145. Ladwig G, Harth A (2011) 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. Corbellini A, Mateos C, Zunino A, Godoy D, Schiaffino S (2017) Persisting big-data: the NoSQL landscape. Inf Syst 63:1–23
147. Barbará D (2002) Requirements for clustering data streams. SIGKDD Explor Newsl 3(2):23–27. https://doi.org/10.1145/507515.507519
148. Gama J, Aguilar-Ruiz J (2007) Knowledge discovery from data streams. Intell Data Anal 11(1):1–2
149. Meir-Huber M, Köhler M (2014) Big data in Austria. Austrian Ministry for Transport, Innovation and Technology (BMVIT), Technical report
150. Nural MV, Peng H, Miller JA (2017) Using meta-learning for model type selection in predictive big data analytics. In: 2017 IEEE international conference on Big Data (Big Data). IEEE, pp 2027–2036
151. Cunha T, Soares C, de Carvalho AC (2018) Metalearning and recommender systems: a literature review and empirical study on the algorithm selection problem for collaborative filtering. Inf Sci 423:128–144
152. Blair G, Bencomo N, France R (2009) Models@ run.time. Computer 42(10):22–27
153. Schmid S, Gerostathopoulos I, Prehofer C, Bures T (2017) Self-adaptation based on big data analytics: a model problem and tool. In: IEEE/ACM 12th international symposium on software engineering for adaptive and self-managing systems (SEAMS). IEEE, pp 102–108
154. Hartmann T, Moawad A, Fouquet F, Nain G, Klein J, Traon YL (2015) 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
155. van der Aalst W, Damiani E (2015) Processes meet big data: connecting data science with process science. IEEE Trans Serv Comput 8(6):810–819
156. Luckham DC (2001) The power of events: an introduction to complex event processing in distributed enterprise systems. Addison-Wesley, Boston
157. van der Aalst WMP (2012) Process mining. Commun ACM 55(8):76–83
158. van der Aalst WMP, Adriansyah A, de Medeiros AKA, Arcieri F, Baier T, Blickle T, Bose RPJC, van den Brand P, Brandtjen R, Buijs JCAM, Burattin A, Carmona J, Castellanos M, Claes J, Cook J, Costantini N, Curbera F, Damiani E, de Leoni M, Delias P, van Dongen BF, Dumas M, Dustdar S, Fahland D, Ferreira DR, Gaaloul W, van Geffen F, Goel S, Günther CW, Guzzo A, Harmon P, ter Hofstede AHM, Hoogland J, Ingvaldsen JE, Kato K, Kuhn R, Kumar A, Rosa ML, Maggi FM, Malerba D, Mans RS, Manuel A, McCreesh M, Mello P, Mendling J, Montali M, Nezhad H R M, zur Muehlen M, Munoz-Gama J, Pontieri L, Ribeiro J, Rozinat A, Pérez HS, Pérez RS, Sepúlveda M, Sinur J, Soffer P, Song M, Sperduti A, Stilo G, Stoel C, Swenson KD, Talamo M, Tan W, Turner C, Vanthienen J, Varvaressos G, Verbeek E, Verdonk M, Vigo R, Wang J, Weber B, Weidlich M, Weijters T, Wen L, Westergaard M, Wynn MT (2011) Process mining manifesto. In: Proceedings of the business process management workshops (BPM). Springer, pp 169–194
159. Dumas M, van der Aalst WMP, ter Hofstede AHM (2005) Process-aware information systems: bridging people and software through process technology. Wiley, Hoboken
160. van Dongen BF, van der Aalst WMP (2005) 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)
161. Al-Ali H, Damiani E, Al-Qutayri M, Abu-Matar M, Mizouni R (2016) Translating bpmn to business rules. In: International symposium on data-driven process discovery and analysis. Springer, pp 22–36
162. Hripcsak G, Rothschild AS (2005) Agreement, the f-measure, and reliability in information retrieval. J Am Med Inform Assoc 12(3):296–298
163. Gilson O, Silva N, Grant PW, Chen M (2008) From web data to visualization via ontology mapping. Coput

Graph Forum 27(3):959–966. https://doi.org/10.1111/j.1467-8659.2008.01230.x

164. Nazemi K, Burkhardt D, Breyer M, Stab C, Fellner DW (2010) 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, pp 163–173

165. Nazemi K, Breyer M, Forster J, Burkhardt D, Kuijper A (2011) Interacting with semantics: a user-centered visualization adaptation based on semantics data. In: Smith MJ, Salvendy G (eds) Human interface and the management of information. Interacting with information. Springer, Berlin, Heidelberg pp 239–248

166. Melo C, Mikheev A, Le-Grand B, Aufaure M-A (2012) Cubix: a visual analytics tool for conceptual and semantic data. In: IEEE 12th international conference on data mining workshops (ICDMW). IEEE, pp 894–897

167. Fluit C, Sabou M, Van Harmelen F (2006) Ontology-Based information visualization: toward semantic web applications. In: Geroimenko V, Chen C (eds) Visualizing the semantic Web: XML-Based internet and information visualization. Springer, London, pp 45–58. https://doi.org/10.1007/1-84628-290-X_3

168. Krivov S, Williams R, Villa F (2007) Growl: a tool for visualization and editing of owl ontologies. Web Semant Sci Serv Agents World Wide Web 5(2):54–57

169. Chu D, Sheets DA, Zhao Y, Wu Y, Yang J, Zheng M, Chen G (2014) Visualizing hidden themes of taxi movement with semantic transformation. In: Visualization symposium (PacificVis), IEEE pacific. IEEE, pp 137–144

170. Catarci T, Scannapieco M, Console M, Demetrescu C (2017) My (fair) big data. In: 2017 IEEE international conference on Big Data, BigData 2017, Boston, MA, USA, December 11–14, pp 2974–2979 [Online]. https://doi.org/10.1109/BigData.2017.8258267

171. Oracle (2015) The five most common big data integration mistakes to avoid, white paper. http://er2016.cs.titech.ac.jp/program/panel.html. Accessed 20 June 2017 [Online]

172. Ali SMF, Wrembel R (2017) From conceptual design to performance optimization of ETL workflows: current state of research and open problems. VLDB J. [Online]. https://doi.org/10.1007/s00778-017-0477-2

173. Olston C, Reed B, Srivastava U, Kumar R, Tomkins A (2008) Pig latin: a not-so-foreign language for data processing. In: Proceedings of the ACM SIGMOD international conference on management of data, SIGMOD, Vancouver, BC, Canada, June 10–12, pp 1099–1110 [Online]. https://doi.org/10.1145/1376616.1376726

174. Venkataraman S, Yang Z, Liu D, Liang E, Falaki H, Meng X, Xin R, Ghodsi A, Franklin MJ, Stoica I, Zaharia M (2016) 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, pp 1099–1104 [Online]. https://doi.org/10.1145/2882903.2903740

175. Dinter B, Gluchowski P, Schieder C (2015) A stakeholder lens on metadata management in business intelligence and big data-results of an empirical investigation

176. Yazici A, George R (1999) Fuzzy database modeling, ser. Studies in fuzziness and soft computing. Physica Verlag, vol 26. iSBN 978-3-7908-1171-1

177. Shafer G (1976) A mathematical theory of evidence. Princeton University Press, Princeton

178. Wanders B, van Keulen M, van der Vet P (2015) Uncertain groupings: probabilistic combination of grouping data. In: Proceedings of DEXA, ser. LNCS, vol 9261. Springer, pp 236–250. https://doi.org/10.1007/978-3-319-22849-5_17

179. Huang J, Antova L, Koch C, Olteanu D (2009) MayBMS: a probabilistic database management system. In: Proceedings of SIGMOD. ACM, pp 1071–1074. https://doi.org/10.1145/1559845.1559984

180. Thiele M, Fischer U, Lehner W (2009) Partition-based workload scheduling in living data warehouse environments. Inf Syst 34(4–5):382–399

181. Angelini M, Santucci G (2013) Modeling incremental visualizations. In: Proceedings of the EuroVis workshop on visual analytics (EuroVA13), pp 13–17

182. Schulz H-J, Angelini M, Santucci G, Schumann H (2016) An enhanced visualization process model for incremental visualization. IEEE Trans Vis Comput Graph 22(7):1830–1842

183. Stolper CD, Perer A, Gotz D (2014) Progressive visual analytics: user-driven visual exploration of in-progress analytics. IEEE Trans Vis Comput Graph 20(12):1653–1662

184. Fekete J-D, Primet R (2016) Progressive analytics: a computation paradigm for exploratory data analysis. arXiv preprint arXiv:1607.05162

185. Shneiderman B, Aris A (2006) Network visualization by semantic substrates. IEEE Trans Vis Comput Graph 12(5):733–740

186. Wu D, Greer MJ, Rosen DW, Schaefer D (2013) Cloud manufacturing: strategic vision and state-of-the-art. J Manuf Syst 32(4):564–579

187. Martin KE (2015) Ethical issues in the big data industry. MIS Q Exec 14:2