World Wide Web (2023) 26:4001–4023
https://doi.org/10.1007/s11280-023-01218-3

A semantic and service-based approach for adaptive mutli-structured data curation in data lakehouses

Firas Zouari[1] · Chirine Ghedira-Guegan[1] · Khouloud Boukadi[2] · Nadia Kabachi[3]

Received: 6 March 2023 / Revised: 17 September 2023 / Accepted: 14 October 2023 /
Published online: 6 November 2023

Abstract

Recently, we noticed the emergence of several data management architectures to cope with the challenges imposed by big data. Among them, data lakehouses are receiving much interest from industrial and academic fields due to their ability to hold disparate multi-structured batch and streaming data sources in a single data repository. Thus, the heterogeneous and complex aspect of the data requires a dedicated process to improve their quality and retrieve value from them. Therefore, data curation encompasses several tasks that clean and enrich data to ensure it continues to fit the user requirements. Nevertheless, most existing data curation approaches need more dynamics, flexibility, and customization in constituting the data curation pipeline to align with end user requirements that may vary according to her/his decision context. Moreover, they are dedicated to curating only a single type of structure of batch data sources (e.g., semi-structured). Considering the changing requirements of the user and the need to build a customized data curation pipeline according to the users and the data source characteristics, we propose a service-based framework for adaptive data curation in data lakehouses that encompasses five modules: data collection, data quality evaluation, data characterization, curation service composition, and data curation. The proposed framework is built upon new data characterization and evaluation modular ontology and a curation service composition approach that we detail in the following paper. The experimental findings validate the contributions' performance in terms of effectiveness and execution time.

Keywords Data curation · Service composition · Machine Learning · Reinforcement Learning · Data Lakehouse

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering 2022*
Guest Editors: Richard Chbeir, Helen Huang, Yannis Manolopoulos and Fabrizio Silvestri .

✉ Firas Zouari
firas.zouari@univ-lyon3.fr

✉ Chirine Ghedira-Guegan
chirine.ghedira-guegan@univ-lyon3.fr

Extended author information available on the last page of the article

1 Introduction

We have witnessed an increasing rise in data generation that has increased the need to propose new solutions adapted to manage this massive amount of data. Accordingly, data lakehouses were proposed as a new solution that can carry this amount of data and constitute a new architecture that outperforms traditional storage systems. To be more precise, it proposes a new architecture combining the flexibility and scalability of a data lake with the data structures and data management capabilities of a data warehouse [1]. Nevertheless, such architecture may encompass a vast amount of data in different structures (i.e., structured, semi-structured, unstructured) collected from internal and external sources, which may contain erroneous, duplicate, and missing data. Hence, such data must be treated via cleaning and enrichment before being exploited for later analysis. For this purpose, data curation is a practical solution that overcomes such problems to promote data use. Data curation is the active management process from the data creation point to extract value from it [2]. Thus, it encompasses several data, metadata, and scheme management tasks. These tasks need to be organized conveniently to constitute an effective curation pipeline. Nevertheless, considering the data lakehouses, they encompass several multi-structured data sources that may differ in quality and have different characteristics, which may require the execution of specific data curation tasks. In addition, end users may have different needs and requirements regarding curation of these data sources according to their decision context. Therefore, data characteristics and the decision context must be considered while constituting the data curation pipeline to curate data conveniently [3]. Moreover, the tasks constituting the pipeline need continuous reorganization to fit the changes according to the evolution of the decision context [4]. Accordingly, the challenge we tackle in this work is the identification of the convenient data curation tasks and their organization regarding the data source characteristics and the decision context to align with user expectations. However, most existing data curation approaches and methods do not consider the aforementioned characteristics. This makes them static regarding these changes that may handicap the decision-making process, which needs to be performed in a timely and effective manner.

Thus, we propose a new service-based framework for adaptive batch and streaming data curation to overcome the limitations of existing approaches. Specifically, our proposed framework groups the main data curation stages, like data collection, quality evaluation, and characterization. Moreover, it relies on reinforcement learning to constitute a curation pipeline from a library of curation services adaptively to user requirements. Thus, our proposed approach has the merit of characterizing and evaluating the data sources using a new ontology for data characterization and evaluation. Then, based on the identified characteristics, it generates the curation service composition scheme according to the functional and non-functional user requirements, including her/his decision context. Hence, our main goal is to perform active data curation that considers the changing requirements and the various data structures grouped in the data lakehouses. The proposed contributions are implemented as modules constituting the layers of our framework, such as data collection, quality control, treatment, and curation layers.

The remainder of the present paper details our proposal through the following sections: Section 2 overviews the related work. Section 3 presents the proposed framework for adaptive data curation, while Section 4 details the contributions constituting the modules of this framework, such as data source characterization, data quality evaluation, and the generation of the curation service composition scheme. Section 5 presents the elaborated experiments that assess the effectiveness of our contributions. Finally, Section 6 concludes the paper and presents some future endeavors.

2 Related work

Data management, generally, and data curation, specifically, have attracted many researchers' attention in the last decade. We identified works addressing, on the one hand, data curation operations and sequencing and, on the other hand, the dynamic orchestration of data preparation. Thus, we examined contributions that constitute semi-automatic curation steps to curate data sources like [5]. This work focuses on curating social data through several curation services combined with crowdsourcing. We also examined [6] that ensure data preparation for structured data source curation via loosely coupled modules. In the same context, we identified other architectures and frameworks like KAYAK [7], which presents several data preparation tasks as Direct Acyclic Graph to constitute pipelines. Also, we investigated [8] describing the use of Vadalog, a Knowledge Graph Management System dedicated to performing data science via several tasks, including data wrangling (i.e., information extraction, stemming, entity resolution, etc.). By analyzing the existing proposals, we noticed that most presented works could not simultaneously treat different data source structures (i.e., unstructured, semi-structured, or structured). Nevertheless, data lakehouses hold different data structures that require sophisticated curating tools. Considering curation process automation, several approaches are only partially automatic. However, the intervention of the human actor needs to be more accurate and timely. Our analysis also investigated the flexibility of the studied approaches. Most proposals showed static behavior regarding the changing decision process features. However, only a few works considered this aspect, and offered only a low adaptivity level to end-user needs. To the best of our knowledge, all the examined approaches treat exclusively batch data sources.

Regarding the limits of the approaches presented, it is necessary to propose a solution that overcomes them by adapting data curation to several users' needs. Specifically, our proposal ensures data curation for batch and streaming data simultaneously and adaptively to users' functional and non-functional requirements, in particular, decision context, user profile, constraints and preferences, and data structure.

3 Adaptive data curation framework

Data lakehouses hold multi-structured data sources ingested in batch and streaming. Hence, data curation is a crucial step to be applied before analyzing data sources, which may enhance the quality of outcome. However, curating these heterogeneous data sources while simultaneously considering several user requirements is challenging. For this purpose, we propose an adaptive data curation framework for batch and streaming data sources that aims to optimize further data analysis steps regarding execution time and alignment with user needs.

As presented in Figure 1, our framework encompasses four layers: data collection, data quality control, data treatment, and data curation layers. Through these layers, the data curation framework ensures adaptivity from the moment of data collection up to the generation of a curation pipeline. Indeed, the data collection layer collects batch and streaming data sources and metadata about streaming data, providers, location, and temporal information.

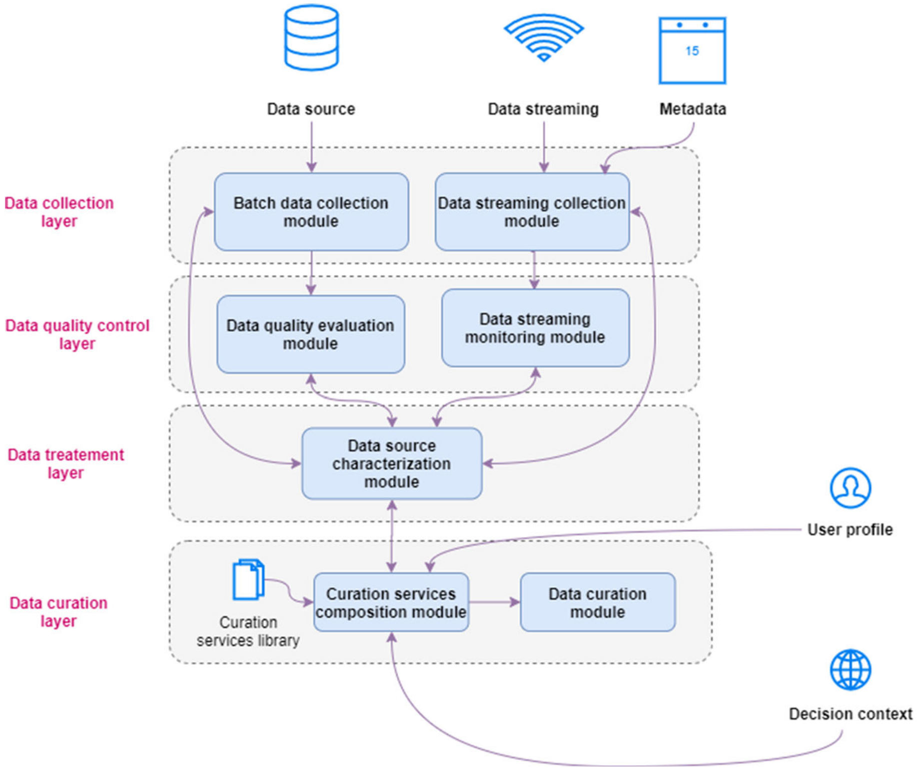

Data source
Data streaming
Metadata
15
Data collection layer
Batch data collection module
Data streaming collection module
Data quality control layer
Data quality evaluation module
Data streaming monitoring module
Data treatement layer
Data source characterization module
User profile
Data curation layer
Curation services library
Curation services composition module
Data curation module
Decision context

Figure 1 Adaptive data curation framework

Then, the framework evaluates data quality using the data quality evaluation and data streaming monitoring modules.

These modules evaluate data quality using quality dimensions, including data accuracy, timeliness, believability, verifiability, and reputation. According to the estimated quality dimension values, the framework decides whether a data curation needs to be executed. Data curation is performed when the data quality dimensions are evaluated below a threshold β. This threshold could be defined and modified by the end user.

After the data evaluation process, the evaluated data quality dimensions are transmitted to the data characterization module that figures in the data treatment layer. The data source characterization module extracts the main characteristics needed for the data curation process, like the format, data source type, and specific data curation tasks (See the following section). Based on these characteristics, the user profile, and the decision context, the data curation layer picks the most suitable curation services from a library of curation services to constitute a customized curation pipeline. Indeed, each curation service performs a curation task like removing duplicate records, anomaly detection, etc. These curation tasks must be appropriately organized to create the pipeline and curate a data source. For this reason, we treat data curation as a service composition problem in which we compose curation services to align with end-user expectations. Thus, the curation framework takes advantage of artificial intelligence mechanisms and, specifically, machine learning and semantic technologies to address the abovementioned challenges related to data source heterogeneity, decision context instability, restriction in terms of execution time, and accuracy of outcomes. Semantic technologies, in turn, have proven to be effective in many cases and can better describe data sources' characteristics and quality that may be involved in constituting curation pipelines (i.e., generating curation service composition schemes).

In addition, machine learning mechanisms could enhance data curation pipeline organization since they automate curation tasks and gain increasing experience. Technically, we implemented the proposed framework as the service-oriented architecture since it is reliable, scalable, and loosely coupled.

The rest of the paper details the steps related to proposing service-based adaptive data curation, such as data characterization, quality evaluation, and the generation of the curation service composition scheme.

4 Towards an adaptive curation service composition method

To achieve the above-mentioned objectives, we propose an adaptive curation service composition method that encompasses the following steps: data quality evaluation, characterization, and generation of the curation service composition scheme. The steps of this method were implemented as a framework, as described in the previous section. We formalize and illustrate our method and the main features of each step in what follows.

4.1 Data quality evaluation

Data quality evaluation is a crucial step that assesses data source quality to decide whether it needs data curation. For this purpose, we propose a new ontology for data quality evaluation and characterization. The modular proposed ontology describes data sources from different perspectives and encompasses four modules: data source description, data quality, provenance, and platform modules, as depicted in Figure 2.

We rely on the ontology's data quality module to design our framework's data quality evaluation layer. This module evaluates the quality of data and data sources via reasoning based on several quality factors, such as quality dimensions, standards, certificates, quality policies, and user quality feedback. Indeed, we adopted and reused standards proposed by W3C, such as [9–13], to design the classes constituting the data quality evaluation module. By investigating the standards presented in [14] and considering the context of the present work, we relied on several data quality dimensions to evaluate the quality of the source and the data from different perspectives.

- Data and Source accuracy: it is primordial to check the data precision and source relevance.
- Time-related accuracy dimensions: the time aspect is crucial for checking the temporal validity of the data, which affects prediction reliability. As data lakehouses keep ingesting raw data sources, the data sources may continue to evolve, and some data may become obsolete. Thus, it is necessary to test whether data are still temporally valid.
- Trustworthiness: as data lakehouses ingest data from several sources with different qualities and to attribute confidence to a data source, we need to check its reputation as a data source as well as the reputation of its provider. We can ask questions like "Can we believe its contents?", "Who is the provider of this data source?", "What is the users' review?"

Figure 3 depicts the core classes of the data quality module. As depicted in this figure, data quality is evaluated based on different factors, such as standards, certificates, user feed-

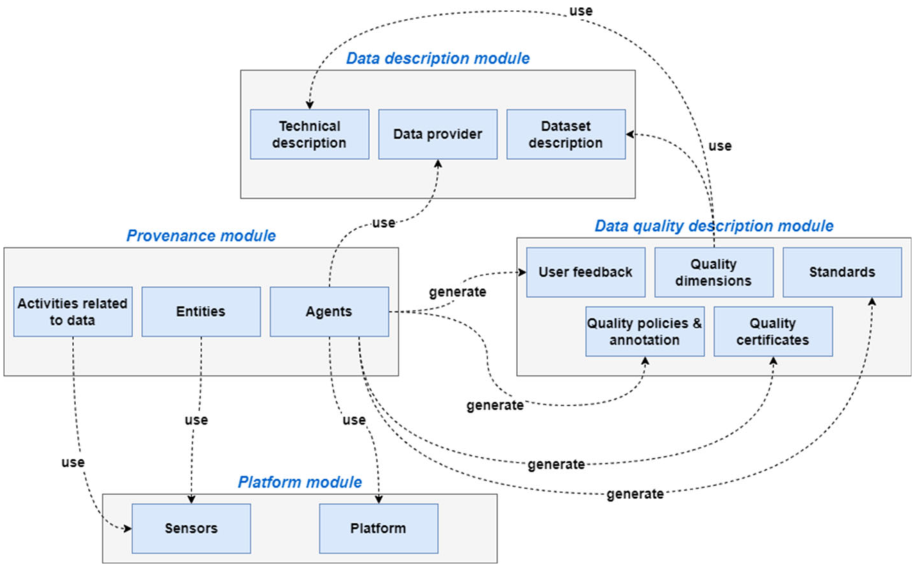


Figure 2 Modular ontology for data source characterization and quality evaluation

back, quality annotations, and the incorporated dimensions for data and data source quality evaluation. Specifically, we focus on data accuracy, currency, volatility, and timeliness dimensions to evaluate data quality. Besides, we rely on source accuracy, verifiability, reputation, and believability to assess data source quality. We defined the following inference rules to compute the values of each quality dimension.

Data accuracy is verified via data quality verification procedures that may be related to the domain. For instance, the quality of data collected from sensors could be assessed via continuous monitoring of data values. In case of sudden changes in the observed values that are out of the normal range, we could identify sensor failure. Source accuracy can be computed based on the accuracy of all data source items. Precisely, source accuracy is the ratio between accurate records and total records. Considering currency, it concerns how promptly

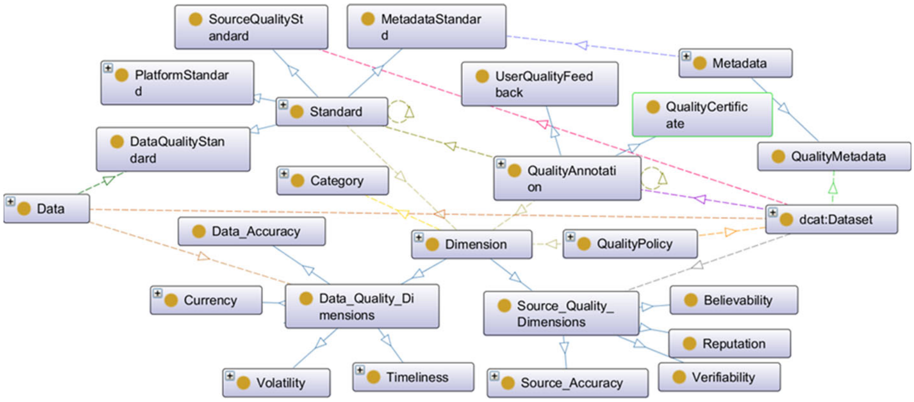


Figure 3 The core classes of the data quality module

data are updated to changes occurring in the real world [15]. Many methods were proposed to measure the currency of a data source. We adopt the following equation, which is widely used.

$$Currency = Age + DeliveryTime - InputTime \tag{1}$$

Where:

- *Age* is the age of the data during data collection
- *DeliveryTime* is the data collection date
- *InputTime* is the data entry date.

These parameters are inferred from the data source's metadata.

We consider volatility as the length of time that data remain valid [15]. Thus, we compute the duration between the current and last modification dates and the accuracy to get the remaining validity period, as described in the following equation.

$$Volatility = Currentdate - (LastModification + accuralPeriod) \tag{2}$$

Where:

- *Currentdate* is the system's current date
- *LastModification* is the date of the last modification on the file
- *accrualPeriod* considers the date from and including the previous distribution date. For instance, some data sources, like reports, may have a regular period for proposing new versions (e.g., monthly reports). Hence, this frequency may impact data volatility.

As for timeliness, it expresses how current the data are for the task at hand [15]. We adopt the following equation to measure the timeliness of a data source:

$$Timeliness = Max(0, 1 - (Currency/Volatility)) \tag{3}$$

Considering trustworthiness, it relies on three dimensions: believability, reputation, and verifiability [15]. We adopted the 7Ws Model [16] that encompasses questions to be answered via queries over the ontology during data collection. The obtained answers are used to infer a score ranging from 0 to 7, indicating low to high trustworthiness. The questions are as follows:

- What are the data?
- Who is the author/organization who created the dataset?
- Why is the dataset created?
- How was it collected - what events led up to its collection?
- When was it collected?
- Where was it collected?
- Which instruments were used to collect it?

These questions detail knowledge related to the provenance of data sources which we describe via the provenance module. The provenance module is based on the logic proposed in [10] and tracks the origins of a data unit. As depicted in Figure 4, the three main concepts of this module are Entity, Activity, and Agent. The Agent class represents the agent who manipulates the activities (e.g., SoftwareAgent, Person, and Organization). An agent may act on behalf of another agent and use an entity or ensure an activity. The Activity class is designed to illustrate the activities leading to generating the data. These activities, in turn, can be associated with agents and use entities. Each activity may have start and end dates and may

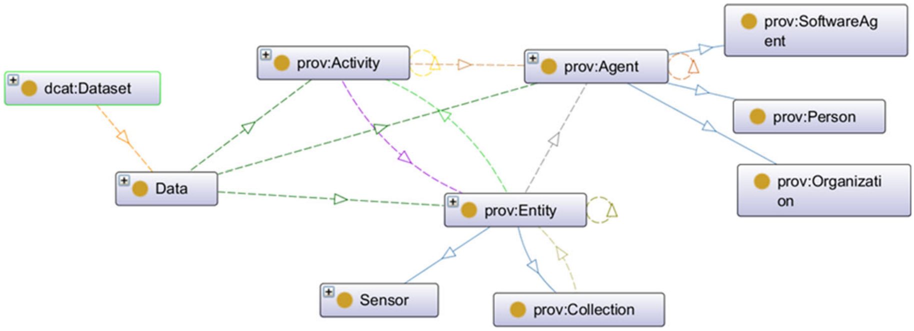


Figure 4 The core classes of the provenance module

transmit information to other activities. As for the class entity, it describes entities dealing with data units (e.g., Sensor). An entity may be derived from another entity or activity. For instance, we have represented the Collection class that groups entities (e.g., the collection "Sensor Network" groups the entities "Sensor"). To illustrate the utility of each class, we assume *agent* Bob is executing the *activity* data analysis, in which it checks and analyzes data collected from the sensor entities. In the following section, we detail the data source characterization module.

4.2 Data source characterization

The data source characterization step extracts the data source's characteristics required for generating a curation service composition scheme. Indeed, the data source characteristics may impact the selection of curation services, which influences, consequently, the composition of the overall services. Accordingly, we rely on the data source description and platform module to characterize data sources. The data source description module provides several kinds of information, like information on data, such as the period and location of observations, the linguistic system, the different forms of data it contains, and information about the provider. Figure 5 depicts the core classes of the data description module. This module also encompasses technical information, such as the data format in which the dataset is provided (Distribution) (e.g., an XML dataset, a plain text file, an SQL database, etc.). The data format may be combined with data properties like the URL to access a MySQL database, for example, the username and the password.

Moreover, the data source description module describes the usage of the dataset, the tool which exploits it and the right statements, and the license to use it. As for data curation, it relies mainly on the information provided in the data properties and classes of the ontology constituting the data source description module. The latter also encompasses other features that may directly impact the curation process, such as the following:

- ***The data source format (structured (S), semi-structured (SS), or unstructured (US))****: this characteristic helps the curation service composition module to select suitable services according to the source type.*
- ***Does the data source include a URL in its data values?****: this characteristic determines the need to invoke the URL extraction service.*
- ***Does the data source need to be converted to another format?****: some data sources may need data format conversion before performing data curation. For instance, a plain text*

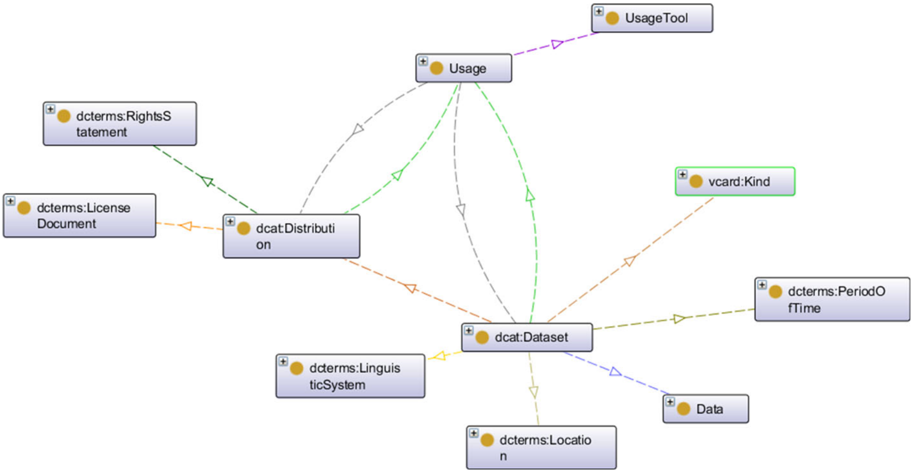


Figure 5 The core classes of the data description module

file needs to be converted into another format (e.g., an XML file) to apply the curation process properly. Thus, this characteristic distinguishes whether the "Converter Service" needs to be invoked.

- ***Does the data source need to undergo a PoS Tagging process?****: some data sources contain paragraphs that need to be enriched via POS Tagging. The latter is a process that describes words with their grammatical information like nouns, verbs, gender, number, etc. This characteristic identifies the need to invoke the POS Tagging process to annotate the data source's paragraphs.*
- ***Is it streaming data?****: this characteristic distinguishes the ingestion mode, like batch and streaming data, to invoke proper curation services for each mode.*

In addition to these characteristics, we track the different actors who can perform data generation to trace the origins of the data. Indeed, we trace the relation between actors, data related activities, and entities via the provenance module that we presented in Section 4.1. As for entities, we design the platform module in our proposed modular ontology that explicitly describes sensor generated data, like real-time data. The platform module describes the different devices from which the observations (e.g., heart rate) were collected, which may host entities like sensors. As depicted in Figure 6, every sensor is characterized by an observable property and a feature of interest. The latter describes the thing whose property is being estimated or figured in the observation to get a result.

Similarly, the observable property class depicts a visible quality of a FeatureOfInterest. Let us take an example to illustrate the role of each class, in which we assume the heart rate of an individual is measured via a smartwatch. Therefore, we can present this information via the class *Smartwatch*, which is the subclass of the class *Platform*. The *Smartwatch* class may encompass an individual named "Apple Watch Series 8" that contains a *Sensor* represented by the individual "photoplethysmography", which is the sensor used to measure heart rate in Apple watches.

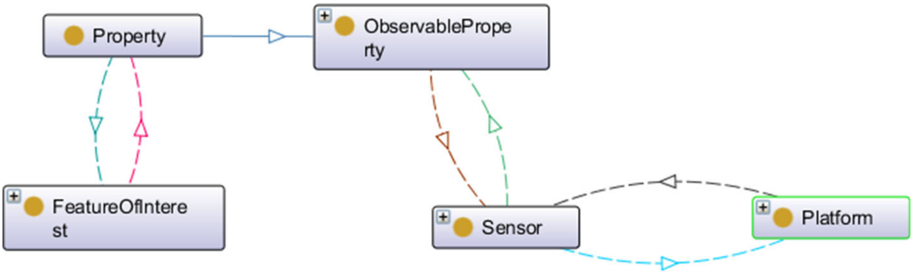


Figure 6 The core classes of the platform module

Following data characterization, we provide the formal description of a data source as follows:

$$\mathbf{D}=< DN, DAtt, Do, MAtt, DCh >$$

where:

- **DN** is the data source name
- **DAtt** represents the data attributes
- **Do** represents the data records
- **MAtt** is the set of attributes taken from a Metadata M
- **DCh** represents the characteristics needed for adaptive data curation that were extracted from the data source via the data description module.

Metadata are defined as:

$$\mathbf{M} =< Mn, MAtt, MVal >$$

where:

- **Mn** is the metadata name
- **MAtt** represents the metadata attributes
- **MVal** represents the data objects

4.3 Curation service composition scheme generation step

In [17] and [18], we illustrated our curation service composition approach with an in-depth discussion of different steps, including data preprocessing. In this paper, our primary focus is on the adaptive identification of the optimal curation service composition scheme. After extracting the data source characteristics, the next step generates the convenient curation service composition scheme, in which each service performs a specific curation task to curate a data source. Following literature analysis, we propose the taxonomy depicted in Figure 7 that presents the main categories of batch and streaming data curation tasks, showing the primitive operations for data curation. We highlight that the concept drift detection task is dedicated exclusively to streaming data curation. This task detects captured streaming data deviation due to a sensor failure, for example. Yet, the other categories' curation tasks are convenient for batch and streaming data curation. We also stress that we relied on this taxonomy to design the library of curation services.

As the curation service composition method employs several curation services, we rely upon and extend the reasoning presented in [19] for service composition, which has proven to be effective. However, the work presented by Wang et al. considers only user preferences and

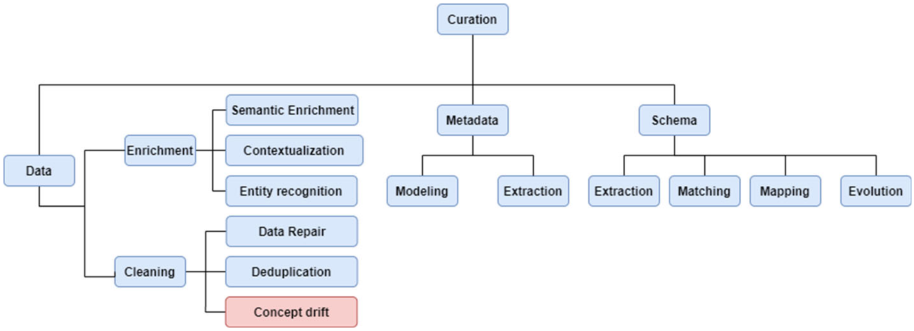


Figure 7 Taxonomy of data curation tasks

quality of services to perform service composition. In the present work, we consider more factors involved in curation service composition, such as the non-functional requirements like user preferences, constraints, decision context, and the quality of services (QoS), as well as the functional requirements like the data structure (i.e., structured, unstructured, semi-structured) and the ingestion mode (i.e., batch or streaming) of the considered data source. For this reason, we relied on reinforcement learning since our proposed method is devoted to dealing with dynamic environments [20].

Thus, as presented in Figure 8, we design the curation step that relies on a training process identifying the optimal policy for composing curation services, followed by a composition process that identifies the convenient services composition scheme using the policy learned. The training process encompasses three tasks: environment initialization, exploration, and exploitation. The initialization task initializes the MDP environment to be explored and exploited by the learning agent, which learns the optimal curation service composition scheme (i.e., a set of transition actions). This scheme is composed of convenient services regarding users' functional and non-functional requirements. Thus, the proposed method relies on the Q-Learning algorithm, which is one of the most effective algorithms for reinforcement

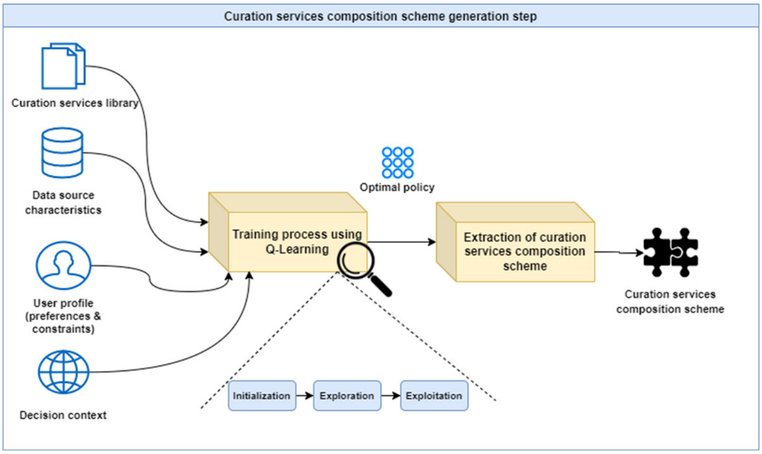

Curation services composition scheme generation step
Curation services library
Data source characteristics
User profile (preferences & constraints)
Decision context
Optimal policy
Training process using Q-Learning
Extraction of curation services composition scheme
Curation services composition scheme
Initialization
Exploration
Exploitation

Figure 8 Illustration of the curation service composition scheme generation step

learning, to identify the optimal curation service composition scheme adaptively, thereby optimizing the execution time and outcome accuracy.

The Q-Learning algorithm is a model-free reinforcement algorithm that employs a learning agent interacting with an environment (usually, a Markov Decision Process) to learn the optimal actions to be taken to carry out a transition from one state to another. By adopting this logic during the learning process, transition weights are learned by the learning agent and assigned to transition actions. These weights represent rewards accumulated after each transition. Hence, we treat the curation service composition as a gain maximization problem that aims to maximize the overall reward.

To be more precise, we represent the curation services in a Markov Decision Process (MDP) containing transition actions that present a curation service. Therefore, we design an MDP environment that includes all the valid possible compositions using curation services and considering all data source types regardless of user requirements.

As some curation services are devoted to curating a single data structure, the environment initializes itself by disabling some actions (i.e., representing curation services) unsuitable for the data source type concerned. Thus, the environment assigns a negative value to the reward to disable inappropriate transition actions. Therefore, the agent will avoid the disabled actions (i.e., that reference inconvenient curation services), since we deal with a gain maximization problem and will select only the transition actions worth a positive reward value. For this purpose, the learning agent employs (4) to compute the transition rewards. This equation involves curation services QoS, user preferences, and constraints (e.g., The QoS response time value >90%) to calculate the reward value. Moreover, the proposed equation may employ several QoS dimensions simultaneously to compute the reward. In addition, it presents user preferences as weights to promote one QoS dimension over another, as depicted in (4). For instance, the end user may be more interested in accuracy than response time. Hence, the learning agent may assign a higher weight for accuracy quality dimension than response time.

We also allow definition of constraints over QoS values, like setting a minimum threshold of QoS M to invoke a curation service. For instance, users may specify a constraint to invoke only services with an accuracy higher than 80%.

Using the quality of services, user preferences, and constraints, the reward function returns a positive value when all the defined users' constraints are fulfilled, otherwise a negative value. Formally, the first part of (4) calculates the difference between constraints defined by the user and the QoS values. Thus, it returns either 1 when all the user constraints are met or -1 otherwise. This equation, in turn, employs (5) to calculate the difference between one QoS dimension and the threshold M set by the user. Then, the value of (5) is normalized to -1 or 1 based on the obtained value. The second part of the equation matches user preferences with the QoS dimensions. Therefore, user preferences are presented as weights multiplied by the QoS dimension values. Subsequently, the reward value is calculated by multiplying the two parts of the equation, which considers user preferences, constraints, and QoS values.

$$R(s) = \underbrace{\frac{\sum_{i=1}^{m} X(i) - 1 + \phi}{\sum_{i=1}^{m} |X(i) - 1 + \phi|}}_{\text{Part1}} * \underbrace{\sum_{k=1}^{m} w_k * D_k}_{\text{Part2}} \tag{4}$$

$$X(k) = \frac{\sum_{i=1}^{m} |D_k - M_k + \phi|}{D_k - M_k + \phi} \tag{5}$$

where:

- **w** represents user preferences regarding a QoS, defined as weight ranging from 0 to 1
- **D** is a normalized value of QoS dimension evaluation ranging from 0 to 1
- **M** represents a minimum threshold set by the user for QoS that needs to be fulfilled to invoke the service. The value of M ranges from 0 to 1
- **Œ** is a normalization value that needs to be strictly higher than 0 and lower than 1

As the reward function relies mainly on the QoS, user preferences, and constraints, we present in what follows the components involved in the training process (See Figure 8) by formally describing the curation services, the library of services, and the user profile. We describe each curation service CS by an ID, a name, quality of service (QoS), and an operation. Thus, a curation service is presented as:

$$\mathbf{CS = < Id, CSN, QoS, Op >}$$

where:

- **Id** represents the curation service Id
- **CSN** is the curation service name
- **QoS** is a set of evaluated QoS dimensions. It contains QoS dimension $\mathbf{QoS_D}$ and the evaluated QoS value $\mathbf{QoS_V}$ presented as pairs and assigned for each assessed QoS dimension
- **Op** is the operation name to be executed following a service invocation.

The user profile groups the user preferences and the group preferences. A user can be a part of a group of users, and each group may have preferences aggregated from individual users' preferences. User group preferences allow individual users to use their group preferences instead of their own to promote one QoS dimension over another. Group preferences also make it possible to save efforts and to learn from other members since they share information about individual preferences between users. Thus, we describe a user profile as:

$$\mathbf{U = < Np, Pru, G >}$$

where:

- **Np** represents the user profile name
- **Pru** is a set that represents the user's preferences regarding a decision context **C**
- **G** represents a group of user profiles. A group is characterized by group name **Ng** and group preferences **Prg** concerning a decision context **C**

We describe the decision context representing the user's surroundings as:

$$\mathbf{C = < Nc,Tc >}$$

where:

- **Nc** represents the name of the context
- **Tc** is the decision context type (e.g., crisis, ordinary situation, etc.). We rely on the proposal in [21] to design the characteristics of the decision context

We implemented the different steps of the curation service composition scheme generation method as modules of the adaptive data curation framework that we detail in the next section.

4.4 Implementation

To illustrate the functioning and deployment of the proposed framework, we relied on the following two scenarios. We assume the adaptive data curation framework is implemented within a crisis management system based on different data management stages, including data curation.

Scenario 1 Let us take Alice, a Deputy Senior Defense and Security Officer at the Ministry of Health, and Bob, an infectious disease specialist, who use this crisis management system to predict and manage health outbreaks. For instance, we specify that Alice uses the system in a critical health situation while Bob uses it in an ordinary case. Thus, their needs regarding the accuracy of outcomes and the system response time could be different. Indeed, response time may be crucial for Alice, while outcome accuracy in Bob's case is less urgent.

We deem they use this system to analyze multi-structured data sources ingested in batch and streaming modes from various providers (e.g., web, sensors, social networks, etc.). In the following example, we focus on data curation, which is part of this crisis management system. We assume that Alice wants to decide on a critical health outbreak using different data sources, including sensors. Therefore, the proposed data curation framework ingests data via the data streaming collection module and monitors the data streams using the data streaming monitoring module and the data quality evaluation ontology. Subsequently, the framework extracts data characteristics using the data source characterization module based on the proposed data characterization ontology. This module identifies several data characteristics, among which the streaming data sources are present in JSON Format. Then, the MDP environment is initialized since the curation service composition module curates streaming semi-structured data by disabling the transition actions referencing improper services, such as batch or structured data curation. Subsequently, the learning agent learns the optimal composition service policy $\pi*$ during the exploration/exploitation process and using (4). Following this, the module composes the curation services that fit Alice's needs using the learned policy $\pi*$ and selecting services with a high response time QoS value. Later, the curation service composition scheme is dispatched to the data curation module to invoke the curation services to curate the data source.

Scenario 2 We took another scenario where Alice uses the crisis management system in an ordinary situation to get some statistics related to infectious diseases. Hence, she has other preferences than the ones presented previously regarding the decision context. Thus, the curation service composition module adapts to the MDP environment and provides another scheme to meet Alice's needs.

As for Bob's case, we assume that he wants to check the last recommendations using the crisis management system to treat a new infectious disease. Therefore, the system ingests data from various sources, such as health institutions, to provide these recommendations. We assume that the data characterization ontology had identified the data sources as databases that require curation services devoted to structured data. Hence, the curation service composition module initializes the MDP environment differently from the previous example (i.e., Alice's case) by promoting curation services for batch and structured data curation according to Bob's preferences in terms of the accuracy of the results. The curation service composition module adapts (4) weights during the exploration/exploitation process to promote the accuracy quality dimension over response time. Accordingly, this module proposes a different curation service composition scheme that fulfills Bob's requirements.

5 Experiments and results

We conducted several experiments to evaluate the core components of our proposed adaptive data curation framework. We relied on a broad spectrum of disease cases using a variety of data sources from the health field. Hence, the following sections detail the evaluation of the performance of (1) our proposed ontology for data characterization and quality evaluation and (2) the performance and (3) the effectiveness of the generation of the data curation service composition schemes.

5.1 Data characterization and quality evaluation ontology

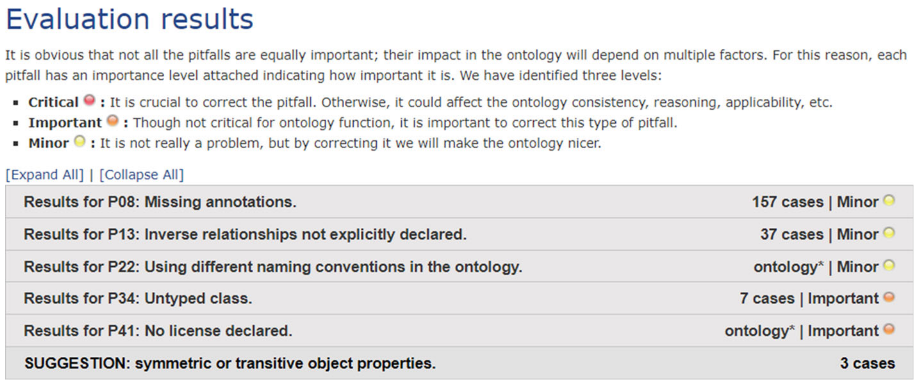
We elaborated experiments to assess data quality evaluation through our proposed ontology for data characterization and quality evaluation. For this purpose, we rely on the Google Mobility dataset[1], a dataset that contains Community Mobility Reports providing insights into the changes in response to policies while combating COVID-19, since it contains dates and periods, allowing us to measure the timeliness and currency quality dimensions. Hence, we aim to measure the effectiveness of the defined inference rules. To do so, we propose a tool that translates the requirements set by the user into SQWRL queries to query the ontology. Then, we use the reasoner PELLET [22] to reason over the ontology to infer each quality dimension value. Thus, we noticed that our ontology successfully computed the value of each data quality dimension using the defined inference rules. For instance, we found that the ontology assigned 100% for the trustworthiness quality dimension since a trustworthy provider provides the dataset.

We also employed the OOPS! Ontology Pitfall Scanner [23], a widely used tool for evaluating ontologies [24–26], which detects anomalies that might exist in the design to assess the quality of the designed ontology. A pitfall is a hidden anomaly that the reasoner may not detect and may lead to later inconsistency. Thus, the OOPS! tool extends the existing pitfall set of existing evaluation approaches and employs a checklist of issues constituting a catalog of pitfalls grouped into different ontology evaluation dimensions: structural, functional, and usability-profiling. Also, the pitfalls are characterized by indicators (e.g., Critical, Important, Minor) that indicate the severity of each pitfall [23].

As depicted in Figure 9, the most commonly detected pitfalls are minor, which do not cause later inconsistency or errors. Most of these pitfalls have occurred due to missing annotations, since we have not yet designed documentation for the ontology users. Considering the important pitfalls (i.e., P34 and P41), the former is caused by using some loaded default libraries of PROTEGE (i.e., the tool we used to design the ontology) that import different classes from external ontologies (e.g., SWRL libraries to create SWRL rules). As for the pitfall P41, it occurs because we still need to publish the ontology on the web. Accordingly, our ontology does not contain critical design errors, which proves our proposal's structural and semantic quality. We elaborated further evaluation of its quality by computing ontology performance metrics.

Thus, such metrics reflect the ontology's functional, analytical, pragmatic, syntactic, cognitive, semantic, social, and practical qualities. For instance, structural quality may impact later activities such as ontology merging and alignment. On the other hand, knowledge quality may measure the extent of the richness of an ontology[27]. Hence, we relied on schema,

[1] https://www.google.com/covid19/mobility/



Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** ● **:** It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ● **:** Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** ○ **:** It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P08: Missing annotations.	**157 cases | Minor** ○
Results for P13: Inverse relationships not explicitly declared.	**37 cases | Minor** ○
Results for P22: Using different naming conventions in the ontology.	**ontology* | Minor** ○
Results for P34: Untyped class.	**7 cases | Important** ●
Results for P41: No license declared.	**ontology* | Important** ●
SUGGESTION: symmetric or transitive object properties.	**3 cases**

Figure 9 Evaluation results using OOPS! Pitfall Scanner

knowledge, and graph evaluation metrics to evaluate the presented dimensions. Tables 1, 2, and 3 depict the evaluation results. Table 1 shows the structural richness of the data characterization and evaluation ontology structure. Indeed, more than one-third of the ontology structure is represented via inheritance and relation shapes, representing the richness of our ontology's structural quality.

In addition to its simplicity (i.e., non-complex ontology), the performance metrics show the knowledge richness depicted via the absolute and maximal breadth. The vast knowledge provided by this ontology is guaranteed via the average population and the class richness values. As mentioned above, the structural and knowledge qualities ensure our ontology's ease of merging, alignment, and reusing.

5.2 Adaptive data curation

The previous experiments focused on the proposed ontology's structural and functional quality. In the following discussion, we further elaborate on the experiments previously outlined in references [17] and [18], where we assessed the effectiveness of our adaptive curation approach and compared its performance to that of reinforcement learning algorithms, specifically First Visit Monte Carlo and Temporal Difference.

These experiments revealed that our approach outperforms these algorithms in terms of scalability, execution time, and alignment with both functional and non-functional requirements. To illustrate, our approach successfully crafted a curation service composition involving 12,000 services, a task that proved unattainable for First Visit Monte Carlo and Temporal Difference.

In this paper, we continue our evaluation of the proposed curation service composition approach. This continuation entails a comprehensive comparison of its performance with

Table 1 Schema evaluation metrics

Inheritance richness	0.30
Relationship richness	0.46
Axiom/Class Ratio	6.15
Class/Relation Ratio	1.76

Table 2 Knowledge metrics

Average population	1.13
Class richness	0.3

existing service composition algorithms, with a specific emphasis on two crucial aspects: (1) scalability and (2) its influence on the data curation process.

5.2.1 Execution time and scalability

We evaluated the performance of our proposed curation service composition approach against service composition benchmarks and baselines that have proven their effectiveness in composing services, such as greedy randomized adaptive search procedure (GRASP) [28], random composition [29], ant colony [30], k nearest neighbors (KNN) [31], and genetic algorithm (GA) [32]. Accordingly, we first proposed a library of curation services consisting of 18 services used for service composition. Then, we simulated the increasing number of services to assess their scalability using 50, 100, 200, 300, 1000, 10000, and 12000 services. We intentionally stopped the experiments when the run time exceeded 1 hour, as we assumed it was immensely time-consuming. Table 4 shows a comparison of the execution time of each service composition method regarding the number of services. As depicted in the Table, the K nearest neighbors, the genetic algorithm, and the GRASP algorithm take too much time to generate a curation service composition scheme. These algorithms take more than 1 hour to create a composition scheme using a library of curation grouping more than 200 services. This huge execution time may be explained by the long training process required to generate a composition scheme.

Nevertheless, when considering data lakehouses, they may contain data ingested in real-time that need curation promptly. Moreover, we stress that the user requirements regarding curation may be unstable and highly changeable. Thus, re-executing a training process to cope with these changes may take too much time and effort. Accordingly, these composition algorithms may not be convenient for this kind of service composition characterized by dynamicity, uncertainty, and a highly changeable environment. We also investigated the performance of the curation service composition using random and ant colony algorithms. Although the random algorithm needs less time to generate a composition scheme, it generates composition schemes randomly that may sometimes be invalid or contain non-convenient services.

Table 3 Graph evaluation metrics

Absolute root cardinality	5
Absolute root node	16
Absolute leaf cardinality	16
Absolute sibling cardinality	16
Maximal depth	2
Absolute breadth	16
Maximal breadth	8
Ratio of leaf fanoutness	0.30
Ratio of sibling fanoutness	0.30
Tangledness	3.31

Table 4 Comparison of the performance of different services composition methods

Number of services	KNN	GA	GRASP	Random	Ant	The proposed method
18 Services	2s	64s	1s	0.4s	1s	0.1s
50 Services	17s	840s	27s	0.5s	1s	0.1s
100 Services	2s	> 1H	420s	1s	5s	0.1s
200 Services	> 1H	-	> 1H	1s	7s	0.1s
300 Services	-	-	-	1s	7s	0.2s
1000 Services	-	-	-	1s	12s	0.2s
10000 Services	-	-	-	120s	420s	5.32s
12000 Services	-	-	-	296s	540s	6.58s

Similarly, the ant colony algorithm showed good performance in terms of execution time. However, both algorithms (i.e., random and ant colony) require more than 2 minutes to generate a composition scheme using a library of services grouping more than 10000 services. As shown in the Table, our curation service composition approach outperforms the examined composition algorithms regarding execution time and scalability since it generates a scheme using more than 10000 services in less than 7 seconds.

5.2.2 Effectiveness of the data curation process

This section presents the experiments elaborated to evaluate the effectiveness of the generated curation service composition schemes. Hence, we invoked the services presented in the composition scheme to examine the curation of different data structures. In other words, we investigated the impact of the selected services for data curation using two structured and unstructured datasets. Specifically, we relied on the same "COVID-19 Community Mobility Reports" dataset in its structured form to conduct the first experiment. We also employed an unstructured dataset[2] containing health news presented as tweets collected from over 15 major health news agencies, such as the BBC, to conduct the second experiment. Given that structured and unstructured data share substantial similarities in their curation needs, we chose to emphasize the composition of services for unstructured and structured data to provide a clear and focused presentation. Indeed, we have implemented numerous curation services that effectively address both unstructured and semi-structured datasets. This inclusivity, while valuable, may potentially obscure the nuances that differentiate these data structures. It's worth noting that we have previously addressed and provided illustrative examples of semi-structured data sources in our prior publications, specifically in papers [17] and [18]. We relied on the data characterization and evaluation ontology to evaluate data source quality and identify the main characteristics needed for the data curation. Then, we evaluated the effectiveness of the data curation process via three-step experiments, namely: (i) generation and verification of a curation service composition scheme, (ii) validation of the outcomes, and (iii) evaluation of the effectiveness of data curation.

Experiment 1 The characterization ontology describes the dataset as structured and requires specific curation tasks dedicated to structured data.

(i) Considering the COVID-19 Community Mobility Reports dataset, the generated curation service composition scheme is as follows: (Metadata extraction service →

[2] https://archive.ics.uci.edu/ml/datasets/Health+News+in+Twitter

Descriptive statistics service → Missing values service → Terminology extraction service → Lexical Service → Rules extraction service → Entity extraction service → Linking service → Synonym service).

(ii) This composition scheme contains services adapted for structured data source curation. Specifically, the metadata extraction and the descriptive statistics services provide statistics about the dataset that may be needed later for data analysis, like the number of features, the size of the dataset, the value ranges, the number of continuous, categorical, and discrete features, and the number of missing values. Figure 10a depicts an extract from the generated metadata and descriptive statistics about missing data.

(iii) Finally, we investigated the effectiveness of the curation services by checking whether the curation process would repair the identified missing data. After invoking the missing data service, we found that our curation framework had successfully filled in the missing data. Nevertheless, fields like "meteo_area" and "place_id" are kept empty because the number of missing data in this field equals the number of rows of the dataset. As for the "iso_3166_2_code" and the "sub_region_2" attributes, they still contain missing values since some regions do not have a second sub-region and may not have an ISO 3166-2 code. Figure 10b depicts the number of missing data by feature after invoking the missing values service.

Then, we monitored the invocation of the next curation services present in the composition scheme (i.e., Terminology extraction service → Lexical Service → Rules extraction service → Entity extraction service → Linking service → Synonym service) constituting the scheme to examine their impact on data source curation.

We stated that the terminology service extracts the dataset features' names to construct a reference model employed by the rules extraction service. The latter identifies rules related to features such as the maximum and the minimum value ranges to detect any semantic violation (i.e., using the extracted rules) and, therefore, check any anomaly that may degrade dataset consistency. Then, the entity extraction service extracts the named entities to be linked with external knowledge bases and enriched via the linking and synonym extraction services. As most rows in this dataset contain numeric values, the role of linking and synonym extraction services is not apparent here. Thus, we emphasize their roles in the second experiment since the employed dataset is unstructured and contains several tweets constituting a set of words that need enrichment.

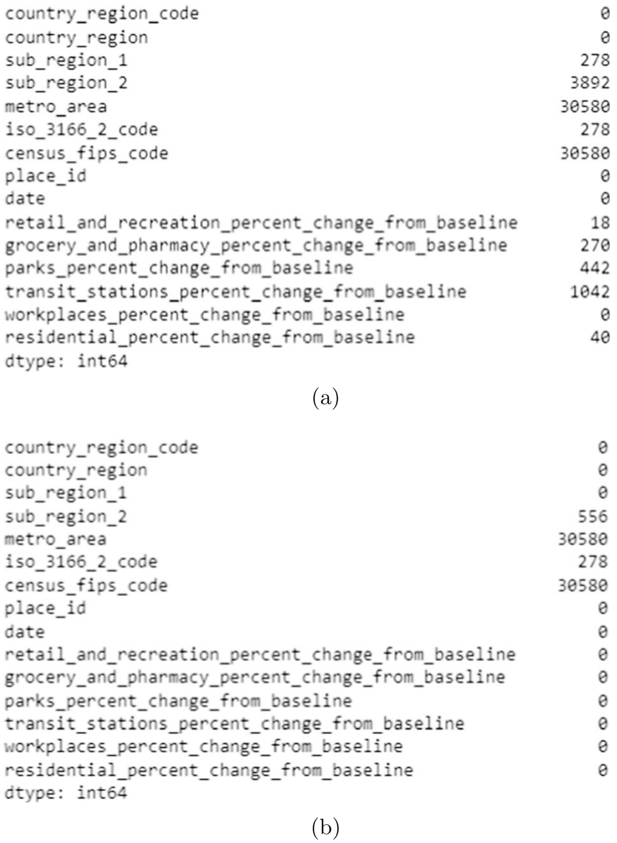
Experiment 2 In this experiment, the data characterization ontology identified data characteristics, such as the unstructured form of the dataset and rows containing URLs.

(i) Based on these characteristics, the curation layer generates the following curation service composition scheme: (URL extraction → Entity extraction service → Linking service → Synonym service).

(ii) We noticed that our framework selected, in this case, only the services devoted to extracting and enriching data, which is appropriate regarding the tweets' characteristics.

(iii) Finally, we investigated the results generated by this composition scheme to evaluate the effectiveness of the enrichment process. Figure 11 illustrates an example of a tweet about cancer and enriched with information extracted from the URL (i.e., the URL that figures in the tweet) and the keywords linked with external knowledge bases. Enrichment is performed via the entity extraction service that described the term "cancer" as a cause of death.

Since the tweet contains an URL from the BBC website, the URL extraction service fetches further information, such as "The international team analysed 77 genes". The linking



```
country_region_code                                  0
country_region                                       0
sub_region_1                                       278
sub_region_2                                      3892
metro_area                                       30580
iso_3166_2_code                                    278
census_fips_code                                 30580
place_id                                             0
date                                                 0
retail_and_recreation_percent_change_from_baseline  18
grocery_and_pharmacy_percent_change_from_baseline  270
parks_percent_change_from_baseline                 442
transit_stations_percent_change_from_baseline     1042
workplaces_percent_change_from_baseline              0
residential_percent_change_from_baseline            40
dtype: int64
```

(a)

```
country_region_code                                  0
country_region                                       0
sub_region_1                                         0
sub_region_2                                       556
metro_area                                       30580
iso_3166_2_code                                    278
census_fips_code                                 30580
place_id                                             0
date                                                 0
retail_and_recreation_percent_change_from_baseline   0
grocery_and_pharmacy_percent_change_from_baseline    0
parks_percent_change_from_baseline                   0
transit_stations_percent_change_from_baseline        0
workplaces_percent_change_from_baseline              0
residential_percent_change_from_baseline             0
dtype: int64
```

(b)

Figure 10 The missing data statistics (a) before and (b) after invocation of the missing data curation service

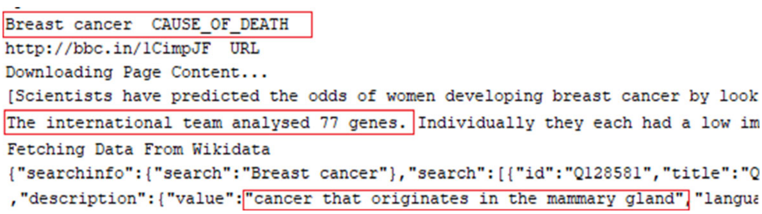
service, in turn, extracts more information, like breast cancer is a "cancer that originates in the mammary gland" from external ontologies. The enriched tweets obtained were validated by a domain expert, proving the effectiveness of the service composition scheme and the appropriate curation. These tweets can thus be used later in data analysis[3] within a prediction model, for example, to forecast cancer cases.

To sum up, the elaborated experiments prove the curation framework's effectiveness in data repair (e.g., replacing missing data) and enrichment by fetching additional information from trusted sources to avoid data noise.

6 Conclusion

We proposed a new framework for adaptive batch and streaming data curation in data lakehouses. The present paper details the core components constituting this framework, such as data characterization, quality evaluation, and data curation. Thus, we offer a new modular

[3] Data analysis is out of the scope of the present paper.



```
Breast cancer  CAUSE_OF_DEATH
http://bbc.in/1CimpJF  URL
Downloading Page Content...
[Scientists have predicted the odds of women developing breast cancer by look
The international team analysed 77 genes. Individually they each had a low im
Fetching Data From Wikidata
{"searchinfo":{"search":"Breast cancer"},"search":[{"id":"Q128581","title":"Q
,"description":{"value":"cancer that originates in the mammary gland","langua
```

Figure 11 An extract from enrichment information for a tweet concerning breast cancer

ontology for data characterization and quality evaluation. It evaluates data source quality to decide whether it needs data curation. In that case, the data characterization module extracts the required characteristics to elaborate the data curation process. Accordingly, our second contribution is a new approach to adaptive generation of curation service composition schemes. This approach takes advantage of reinforcement learning to constitute the core adaptive component of our curation framework. The latter generates the convenient services composition scheme according to users' functional and non-functional requirements, including the data source type, decision context, QoS values, and users' preferences and constraints. We conducted several experiments to evaluate the performance of the proposed data characterization and quality evaluation and the curation service composition approach. The experiments proved the proposed ontology's semantic, reasoning, and structural quality. Moreover, they showed the high performance of our services composition approach in terms of execution time, scalability, and effectiveness. In future works, we plan to investigate further the impact of the curation process on data analysis, and more particularly, we intend to examine the sense and completeness of the proposed enrichment.

Acknowledgements The authors would like to thank Dr. Fatma Guermazi, an oncology physician at Léon-Bérard Center, Lyon, France, for agreeing to validate the effectiveness of the medical enrichment proposed by our data curation approach.

Author Contributions F. Z. and C.G.G. wrote and reviewed the main manuscript text, and K.B. and N.K. participated in elaborating the scientific contribution.

Funding No funding.

Availability of data and materials This declaration is not applicable.

Declarations

Competing interests The authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Ethical Approval This declaration is not applicable.

References

1. Hlupić, T., Oreščanin, D., Ružak, D., Baranović, M.: An overview of current data lake architecture models. pp. 1082–1087 (2022) https://doi.org/10.23919/MIPRO55190.2022.9803717
2. Lord, P., Macdonald, A., Lyon, L., Giaretta, D.: From data deluge to data curation. In: In Proc 3th UK e-Science All Hands Meeting. pp. 371–375 (2004)

3. Akoka, J., Comyn-Wattiau, I., Laoufi, N.: Research on Big Data - A systematic mapping study. Computer Standards and Interfaces. **54**, 105–115 (2017)
4. Tempini, N.: Data curation-research: Practices of data standardization and exploration in a precision medicine database. New Genet. Soc. **40** (2020)
5. Beheshti, A., Vaghani, K., Benatallah, B., Tabebordbar, A.: Crowdcorrect: A curation pipeline for social data cleansing and curation. Inf. Syst. Big Data Era, 24–38 (2018)
6. Konstantinou, N., Abel, E., Bellomarini, L., Bogatu, A., Civili, C., Irfanie, E., Koehler, M., Mazilu, L., Sallinger, E., Fernandes, A.A.A., Gottlob, G., Keane, J.A., Paton, N.W.: VADA: an architecture for end user informed data preparation. J Big Data. **6**(1), 1–32 (2019)
7. Maccioni, A., Torlone, R.: Kayak: A framework for just-in-time data preparation in a data lake. Adv. Inform. Syst. Eng. 474–489 (2018)
8. Bellomarini, L., Fayzrakhmanov, R.R., Gottlob, G., Kravchenko, A., Laurenza, E., Nenov, Y., Reissfelder, S., Sallinger, E., Sherkhonov, E., Vahdati, S., Wu, L.: Data science with vadalog: Knowledge graphs with machine learning and reasoning in practice. Futur. Gener. Comput. Syst. **129**, 407–422 (2022)
9. Debattista, J., Lange, C., Auer, S.: daq, an ontology for dataset quality information. CEUR Workshop Proceedings. pp. 1184 (2014)
10. Lebo, T., Sahoo, S., Mcguinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., Zhao, J.: PROV-O: The PROV Ontology. (2013)
11. Liu, Z., Xu, Z., Xia, X.: Towards systematic analysis and summary of duv-based dataset usage information. pp. 169–172 (2016) https://doi.org/10.1109/WISA.2016.42
12. Shin, D., Lee, S., Kang, J., Park, E.: Data catalogue standards based on dcat for transportation data: Dcat-trans. Journal of Korean Society of Transportation. **37**, 430–444 (2019). https://doi.org/10.7470/jkst.2019.37.5.430
13. Haller, A., Janowicz, K., Cox, S., Phuoc, D., Taylor, K., Lefrançois, M.: Semantic Sensor Network Ontology. (2017)
14. Albertoni, R., Isaac, A.: Introducing the data quality vocabulary (dqv). Semantic Web. **12**,(2020). https://doi.org/10.3233/SW-200382
15. Batini, C., Scannapieco, M.: Erratum to: Data and Information Quality: Dimensions, Principles and Techniques, pp. 1–1 (2016). https://doi.org/10.1007/978-3-319-24106-7_15
16. Walker, J., Frank, M., Thompson, N.: User centred methods for measuring the value of open data. (2015)
17. Zouari, F., Ghedira, C., Kabachi, N., Boukadi, K.: Towards an adaptive curation services composition based on machine learning. IEEE International Conference on Web Services (ICWS), 73–78 (2021)
18. Zouari, F., Ghedira, C., Kabachi, N., Boukadi, K.: A service-based framework for adaptive data curation in data lakehouses. IEEE International Conference on Web Services (ICWS). (2022)
19. Wang, H., Zhou, X., Zhou, X., Liu, W., Li, W., Bouguettaya, A.: Adaptive service composition based on reinforcement learning. Lecture Notes in Computer Science. **6470 LNCS** (60673175), 92–107 (2010)
20. Szepesvári, C.: Algorithms for Reinforcement Learning **9**, 1–89 (2010)
21. Lauras, M., Truptil, S., Bénaben, F.: Towards a better management of complex emergencies through crisis management meta-modelling. Disasters **39**(4), 687–714 (2015)
22. Sirin, E., Parsia, B.: Pellet: An owl dl reasoner. Description Logics, 212–213 (2004)
23. Poveda-Villalón, M., Gomez-Perez, A., Suárez-Figueroa, M.C.: Oops!: A pitfall-based system for ontology diagnosis, 120–148 (2018) https://doi.org/10.4018/978-1-5225-5042-6.ch005
24. Debnath, N.C., Patel, A., Mazumder, D., Manh, P.N., Minh, N.H.: Evaluation of covid-19 ontologies through ontometrics and oops! tools, 351–365 (2022)
25. Alkhariji, L., De, S., Rana, O., Perera, C.: Semantics-based privacy by design for internet of things applications. Futur. Gener. Comput. Syst. **138**, 280–295 (2023). https://doi.org/10.1016/j.future.2022.08.013
26. Yahya, M., Zhou, B., Zheng, Z., Zhou, D., Breslin, J.G., Ali, M.I., Kharlamov, E.: Towards generalized welding ontology in line with iso and knowledge graph construction, 83–88 (2022)
27. Lourdusamy, R., John, A.: A review on metrics for ontology evaluation. 2018 2nd International Conference on Inventive Systems and Control (ICISC), 1415–1421 (2018)
28. Parejo, J., Segura, S., Fernandez, P., Ruiz-Cortés, A.: Qos-aware web services composition using grasp with path relinking. Expert Syst. Appl. **41**, 4211–4223 (2014). https://doi.org/10.1016/j.eswa.2013.12.036
29. Gao, H., Huang, W., Duan, Y.: The cloud-edge-based dynamic reconfiguration to service workflow for mobile ecommerce environments: A qos prediction perspective. ACM Trans. Internet Technol. **21**, 1–23 (2021). https://doi.org/10.1145/3391198
30. Zhang, W., Chang, C.K., Feng, T., Jiang, H.-y.: Qos-based dynamic web service composition with ant colony optimization, 493–502 (2010) https://doi.org/10.1109/COMPSAC.2010.76

31. Raj, T.F.M., Sivapragasam, P., Balakrishnan, R., Lalithambal, G., Ragasubha, S.: Qos based classification using k-nearest neighbor algorithm for effective web service selection. 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 1–4 (2015)
32. Canfora, G., Di Penta, M., Esposito, R., Villani, M.L.: An approach for qos-aware service composition based on genetic algorithms. GECCO 2005-Genetic and Evolutionary Computation Conference. 3387 (2005) https://doi.org/10.1145/1068009.1068189

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Firas Zouari[1] · Chirine Ghedira-Guegan[1] · Khouloud Boukadi[2] · Nadia Kabachi[3]

Khouloud Boukadi
khouloud.boukadi@fsegs.usf.tn

Nadia Kabachi
nadia.kabachi@univ-lyon1.fr

[1] Univ Lyon, Université Jean-Moulin Lyon 3, LIRIS UMR5205, iaelyon School of Management, Lyon, France

[2] University of Sfax, Sfax, Tunisia

[3] Univ Lyon, Univ Lyon 1, UR ERIC and UR4129 P2S - Laboratory “Health, Systemic, Process”, Lyon, France