A Service-Based Framework for Adaptive Data Curation in Data Lakehouses

Firas Zouari[1(✉)], Chirine Ghedira-Guegan[1], Khouloud Boukadi[3], and Nadia Kabachi[2]

[1] Univ Lyon, Université Jean-Moulin Lyon 3, LIRIS UMR5205, iaelyon School of Management, Lyon, France
firas.zouari@univ-lyon3.fr
[2] Univ Lyon, Université Claude Bernard Lyon 1, ERIC UR 3083, Lyon, France
[3] University of Sfax, Sfax, Tunisia

Abstract. Data lakehouses are novel data management designs intended to hold disparate batch and streaming data sources in a single data repository. These data sources could be retrieved from different sources, including sensors, social networks, and open data. Because the data carried in data lakehouses is heterogeneous and complicated, data curation is required to improve its quality. Most existing data curation systems are static, require expert intervention, which can be error-prone and time-consuming, do not meet user expectations, and do not treat real-time data. Given these constraints, 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 services composition, and data curation. The curation services composition module, which leverages several curation services to curate multi-structured batch and streaming data sources, is the focus of this work. A reinforcement learning-based method is provided for adaptively extracting the curation services composition scheme based on the data source type and the end user's functional and non-functional requirements. The experimental findings validate the proposal's effectiveness and demonstrate that it outperforms the First Visit Monte Carlo and Temporal Learning algorithms in terms of scalability, execution time, and alignment with functional and non-functional requirements.

Keywords: Data curation · Service composition · Machine learning · Reinforcement learning · Data lakehouse

1 Introduction

Big data gave birth to diverse concepts such as cloud computing, smart services, data lakes, and most recently, data lakehouses to carry and manage this amount of data. Data lakehouses are data management architectures that present a new generation of unified data platforms. They combine the strengths of data warehouses and data lakes and can hold a variety of data structures (i.e., structured,

R. Chbeir et al. (Eds.): WISE 2022, LNCS 13724, pp. 225–240, 2022.
https://doi.org/10.1007/978-3-031-20891-1_17

semi-structured, or unstructured), as well as batch and streaming data in a well-organized way using metadata. Data lakehouses can be an effective solution in today's urgent contexts and situations, such as crises, that require real-time data collection and analysis. Nevertheless, data heterogeneity and complexity remain critical challenges for data lakehouses. Thus, data cleaning is necessary to enhance data quality before performing data analysis or visualization. For this purpose, data curation ensures managing and promoting data use from its point of creation by enriching or updating it to keep it fit for a specific purpose [7]. It provides more information about the provenance of the data, the original context of measurement and use, and the object of observation to facilitate the re-use of the data [6]. However, the existing data curation approaches are no longer sufficient to curate data in multi-structured data lakehouses and collected from multiple sources (i.e., batch and streaming data) [13]. Besides, the data curation process may be affected by factors such as the data source characteristics and the decision context. Indeed, critical decision contexts, such as crises, are generally evolving and impose restrictions on the execution time and the accuracy of information system outcomes. Therefore, it is paramount to consider the data characteristics and usage context to identify and perform the suitable data curation [1]. Besides, the value of data is never settled, as its semantics are continuously changing, which forces the data curation process to be re-arranged and changed over time [11]. Hence, the data curation needs to be aware of the changing decision process features to optimize the quality of the decision process outcomes, its execution time, and to align with user expectations. We deem *it is challenging to identify the convenient data curation tasks and rearrange them regarding the data source characteristics, the decision context, and the user expectation.* Yet, most existing data curation approaches are static and do not consider the abovementioned features. The latter are handicapping the decision-makers (i.e., those dealing with critical situations) who want to make decisions in a timely and effective manner. Besides, some existing approaches require human intervention, which can be time-consuming and error-prone. Accordingly, we propose a new adaptive data curation framework for batch and streaming data sources to overcome these limitations. The proposed global framework is service-based and encompasses the data curation stages (i.e., data source collection, data quality evaluation, data source characterization, and data curation) as modules. The data curation process encompasses curation services composition and data curation modules that employ a library of curation services in which each service presents a curation task. The curation service composition module composes the curation services adaptively to the decision process features to optimize the data curation process in terms of execution time and alignment to user needs. Subsequently, the data curation module invokes the composing services. To compose curation services, our original contribution relies on artificial intelligence techniques, particularly machine learning, for adaptive generation of data curation services composition scheme according to functional requirements such as the data source characteristics and non-functional requirements like the user preferences and constraints and the decision context. Mainly,

our proposed framework takes advantage of reinforcement learning as a practical solution that can learn over time to make increasingly effective decisions in a dynamic environment like the one of the data lakehouse. This paper presents an overview of the adaptive data curation framework, focusing on the curation services composition module. The remainder of this paper is structured as follows: Sect. 2 presents the related work. Section 3 overviews the proposed framework. Section 4 details the data source characterization and curation services composition modules. Section 5 presents the elaborated experiments and the obtained results. Finally, Sect. 6 concludes the paper and presents some future endeavors.

2 Related Work

Data management in general and data curation, in particular, has attracted the attention of many researchers. Several works in the literature addressing the data curation process exist. Some of them target the sequencing and the dynamic orchestration of data preparation and cleansing automatically combined with semi-automated steps of curation such in [2] for social data via a pool of services, and in [4] which is dedicated only to structured data source curation via loosely coupled data preparation components. In the same vein, some architectures and frameworks were proposed, such as KAYAK [8] that lies between users/applications and the file system (i.e., data storage location), and exposes a set of primitives and tasks for data preparation represented as a Direct Acyclic Graph. Also, [3] details the use of Vadalog, which is a Knowledge Graph Management System dedicated to data science tasks such as data wrangling (i.e., information extraction, stemming, entity resolution, etc.). By analyzing the existing works, we noticed that most proposed approaches could not be generalized to treat at the same time all data source structures (i.e., unstructured, semi-structured, or structured). Yet, data lakehouses contain various data source formats, which require sophisticated tools to curate. Regarding curation process automation, several approaches are not fully automatic. However, the intervention of the human actor is error-prone and time-consuming. Besides, we investigated the flexibility of the studied approaches. Most of them are static regarding the decision process features, and if considered, they ensure a low level of adaptation to end-user needs. Finally, and to our knowledge, all the examined approaches consider only batch data sources.

Considering the above approaches' limits, it is essential to propose a solution that aims to overcome them by performing data curation adaptively. Specifically, our solution should perform data curation for batch and streaming data simultaneously, adaptively to the decision context, the user profile, constraints and preferences, and the type of the considered data source.

3 Adaptive Data Curation Framework

Data lakehouses contain multi-structured data stored as batch and streaming data sources. Hence, data curation is an essential step that needs to be applied

before analyzing data sources to enhance the quality of outcomes. Curating these data sources while simultaneously considering heterogeneous batch and streaming data is challenging. To address this challenge, we propose an adaptive data curation framework for batch and streaming data sources to optimize the further data analysis steps in terms of execution time and alignment with user needs. As depicted in Fig. 1, our framework encompasses the following four layers: data collection, data quality control, data treatment, and data curation layers. Our framework ensures adaptability from the moment of data collection. Indeed, the data collection layer ingests batch and streaming data sources and information about streaming data, data providers, location, and temporal information as metadata. Then, the framework assesses the quality of batch and streaming data via the data quality evaluation module and data streaming monitoring module. The data quality module assesses the data quality and the data source's quality dimensions, such as data accuracy, timeliness, believability, verifiability, and reputation. According to the evaluated quality dimension values, the data curation framework judges whether the data source needs to be curated or not. Specifically, data curation is performed for each data source and follows the data evaluation process when one of the evaluated data quality dimensions is below a threshold β. This threshold can be fixed and modified by the user. Following the data evaluation process, the values of the data quality dimensions and the data source are transmitted to the data characterization module, which we define in the data treatment layer. The data source characterization module extracts the data source characteristics required for the data curation process, like its format, data source type, and specific data curation tasks (See Sect. 4). Based on the extracted features, the user profile, and the decision context, the data curation layer selects the most convenient curation services from a library of curation services to perform a data curation process. Indeed, each curation service ensures a curation task (e.g., removing duplicate records, anomaly detection, etc.). These curation tasks could be combined in a specific way to curate a data source. Hence, we propose to execute the data curation tasks as a service composition while aligning with end-user expectations. Our framework relies on artificial intelligence mechanisms and, more precisely, machine learning techniques to address the challenges mentioned above related to data source heterogeneity, decision context instability, restriction in terms of execution time, and the accuracy of outcomes. Machine learning algorithms can automate curation tasks organization as well as gain experience as they improve accuracy and efficiency to make better decisions. Thus, the curation services composition is enhanced as the learning algorithms gain experience each time. Technically, we deploy our framework following the service-oriented architecture since it is a reliable, scalable, and loosely coupled architecture. The rest of the paper sheds light on the service composition for adaptive data curation.

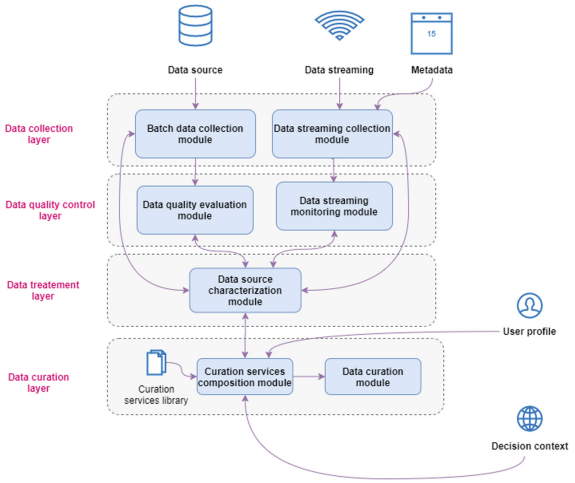


Fig. 1. Adaptive data curation framework

4 Towards an Adaptive Curation Service Composition Method

To attempt our aforementioned objectives, we propose an adaptive curation service composition method that encompasses two steps, namely data characterization and curation services composition scheme generation. We formalize and illustrate our method and the main features of each step in what follows.

4.1 Data Source Characterization Step

The data source characterization step extracts the data source's characteristics required for the curation services composition scheme generation step. Indeed, the data source characteristics may impact the curation services selection, which influences, consequently, the composition of the overall services. We assume a particular data source includes URLs within its attributes. Accordingly, a service named URL extraction service needs to be invoked to fetch information from the data present in the web pages. We formally define a data source as:

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

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 extracted data source characteristics needed for adaptive data curation. The following features characterize a data source:

- ***The data source format (structured (S), semi-structured (SS), or unstructured (US)):*** *This feature guides the service composition module to select the suitable curation services according to the type of the source.*
- ***Does the data source include a URL in its data values ?:*** *This feature helps determine whether the URL extraction service should be invoked.*
- ***Does the data source need to be converted to another format ?:*** *Some data sources require data format conversion before being curated. For example, a plain text file that presents an unstructured data source could be converted into a semi-structured data source (e.g., an XML file) to enhance the curation process. Using this feature, we can distinguish whether the data source needs conversion via the "Converter Service" invocation.*
- ***Does the data source need to undergo a PoS Tagging process?:*** *Some data sources contain paragraphs that need to be annotated via a POS Tagging to enrich them semantically. Indeed, POS Tagging is the association of words in a text with the corresponding grammatical information, such as the gender. Hence, this feature allows identifying whether the data source contains paragraphs that need to be annotated via the POS Tagging process.*
- ***Is it streaming data ?:*** *This feature identifies batch and streaming data to invoke the convenient curation services for each data type.*

A Metadata is defined as :

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

where:

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

Illustration. We assume a JSON dataset named DSPop, which includes demographic information. This dataset also consists of an attribute that contains links to personal web pages.

This dataset can be represented as D = <"DSPop", ("Name", "Age", "Location", " Personal webpage"), {("Alice", 25, "USA", "http://..."), ...}, {("Author", "Charlie"), ("Creation Date", "25/03 /2020"), ("Format", "JSON"), ("Publisher", "Eve")}, ("SS", True, False, False, False)>.

The dataset D is linked with a Metadata M presented as M = <("Author", "Charlie"), ("Creation Date", "25/03/2020"), ("Format", "JSON"), ("Publisher", "Eve")>.

4.2 Curation Services Composition Scheme Generation Step

Following data characterization, the next step generates the convenient curation services composition scheme. Each curation service ensures a specific curation task. The curation tasks are combined to perform a curation process. By analyzing the existing curation works, we propose the taxonomy of the main batch and streaming data curation task categories, depicted in Fig. 2. These curation tasks ensure the necessary operations for data curation. We point out that concept drift detection is devoted to streaming data curation. Concept drift tasks detect the deviation of captured streaming data due to a sensor failure. Yet, the curation tasks within the other categories could be applicable for both batch and streaming data. It is worth noting that we considered this taxonomy when designing the curation services library.

As our proposed method relies on several curation services, we extend the reasoning proposed in [12] for service composition, which has proven to be effective. Yet, the work presented by Wang et al. takes into account only user preferences and QoS to perform service composition. In the proposed work, we consider the various factors implied in curation services composition, namely the non-functional requirements such as user preferences, constraints, and QoS, as well as the functional requirements like the structure (i.e., structured, unstructured, etc.) and the type (i.e., batch or streaming) of the treated data source. Specifically, we adopt reinforcement learning since our proposed method is designed to deal with dynamic environments [10].

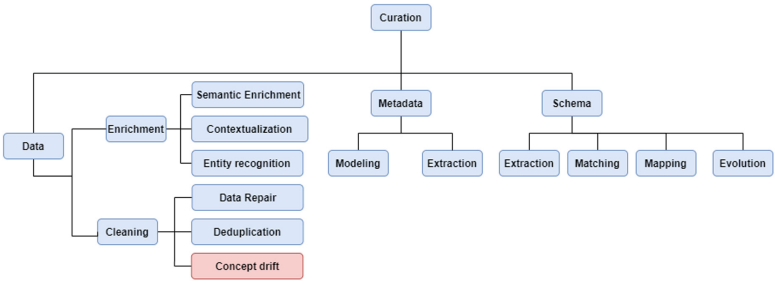


Fig. 2. Taxonomy of data curation tasks.

Thus, and as depicted in Fig. 3, the first process is a training one that identifies the optimal policy to compose curation services. In contrast, the composition process extracts the curation services composition scheme using the generated policy. The training process, in its turn, relies on three tasks: environment initialization, exploration, and exploitation. The initialization task prepares the environment to be explored/exploited by the learning agent to identify the optimal curation services composition scheme (i.e., a set of transition actions). This scheme is generated according to the user's functional and non-functional requirements. To do so, the proposed method uses the Q-Learning

algorithm, one of the most popular algorithms for reinforcement learning, to learn the optimal curation services composition scheme adaptively, which optimizes the execution time and outcomes accuracy. The Q-Learning algorithm is a model-free reinforcement algorithm that defines an agent interacting with an environment (usually, a Markov Decision Process) to learn the optimal actions to be taken for the transition from one state to another. Following this logic, we assign weights to actions to guide the learning agent during the learning process. The weights present rewards accumulated at each transition. Hence, we treat the curation service composition as a gain maximization problem.

We represent the curation services in a Markov Decision Process (MDP) in which each transition action presents a curation service. Thus, in the MDP environment, we present all the valid possible compositions of all the curation services for all data source types regardless of user requirements and environmental factors. As curation services are devoted to curating either semi-structured, unstructured, or structured data sources, the environment adapts itself during the initialization task by disabling some actions (i.e., representing curation services) that are not convenient for the treated data source type. For this purpose, the environment initializes the reward returned by the disabled actions to a negative value. Since we deal with a gain maximization problem, the agent will avoid these actions and select only the transition actions worthing a positive reward value. We propose Eq. 1 to compute the transition rewards. The proposed equation relies on curation services QoS, user preferences, and constraints (e.g., The QoS response time value >90%) to compute the reward value. Equation 1 supports the use of several QoS dimensions together to compute the reward. Regarding user preferences, they could be defined as weights to promote a QoS dimension over another. For example, a user may be more interested in accuracy than response time. Thus, the accuracy quality dimension may receive a higher weight than the response time. We note that it is possible to define constraints over QoS values by setting a minimum threshold M that should be satisfied to invoke a service. For instance, a user may invoke only services with an accuracy quality dimension higher than 80%. By considering the QoS, user preferences, and constraints, the reward function (i.e., Eq. 1) returns a positive value when all users' constraints are satisfied. Otherwise, the function returns a negative value. Since the service composition is a gain maximization problem, the negative rewards prevent the agent from choosing curation services that do not fit the user's constraints. The first part of Eq. 1 computes the difference between user-imposed constraints and QoS values. It returns either 1 if all user constraints are fulfilled or -1 otherwise. Equation 1 relies on Eq. 2 to compute the difference between one QoS dimension and the threshold M defined by the user. Subsequently, the value of Eq. 2 is normalized to -1 or 1 according to the obtained value. The second part allows assigning user preferences to QoS dimensions. Therefore, the preferences are defined as weights to multiply the evaluated QoS dimensions' values. Afterward, the multiplication of the two parts of the equation returns the reward value according to user preferences, constraints, and QoS values.

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

$$X(k) = \frac{\sum_{i=1}^{m} |D_k - M_k + \phi|}{D_k - M_k + \phi} \tag{2}$$

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.
- ϕ 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 illustrate the inputs of the training process (See Fig. 3), which contain these characteristics. Specifically, we formally describe the curation services, the library of services, and the user profile. Each curation service CS is characterized by an ID, a name, its quality (QoS), and an operation. We define a curation service 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 couples and assigned for each assessed QoS dimension
- **Op** is the operation name to be executed following a service invocation.

The user profile encompasses the user preferences and the user group preferences. A user can be a part of a group of users. Each group of users can have group preferences aggregated from users' preferences. During the training process, the user can use either his own preferences or his group preferences to promote a QoS dimension over another. Indeed, group preferences allow sharing information about specified preferences between the users, saving efforts, and learning from other members. Thus, we define 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 define the decision context that represents the user's surroundings as:

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

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 [5] to design the characteristics of the decision context.

We implement the different steps of the curation services composition scheme generation method as modules of the adaptive data curation framework.

Illustration. To illustrate the idea behind the proposed method, we assume that the adaptive data curation framework is implemented in a crisis management system that relies on different data management stages, including data curation. We suppose that Alice, Deputy Senior Defense and Security Officer at the Ministry of Health, and Bob, an infectious disease specialist, use this system to predict and manage health crises. Assume that Alice uses the system in a crisis context, while Bob uses it in an ordinary situation. Hence, they may have different needs regarding the accuracy of outcomes and the system's response time. Indeed, response time may be significant for Alice, while it is not as important as outcomes accuracy in Bob's case. We assume they use this system to deal with multi-structured data sources ingested in batch and streaming modes from different providers (e.g., web, sensors, social networks, etc.). Considering this crisis management system, we focus, in the following example, on data curation. We suppose that Alice wants to decide on a critical health crisis using various data sources, including sensors. To curate the data streams, our proposed data curation framework collects data using the data streaming collection module and monitors the data streams via the data streaming monitoring module. Subsequently, the framework extracts data characteristics using the data source characterization module. This latter identifies several characteristics, among which data are collected from streaming sources in JSON Format. Since the curation service composition module deals with streaming semi-structured data, it initializes the MDP environment by disabling the transition actions referencing improper services, such as the ones for batch or structured data curation. Then, during the exploration/exploitation process, the learning agent learns the optimal composition service policy π^* using Eq. 1. Subsequently, using the learned policy π^*, the module composes the curation services that fit Alice's needs by selecting services with a high response time QoS value. Later, the curation service composition scheme is transmitted to the data curation module to invoke the curation services and perform data curation. In another context, we assume that Alice uses the system in an ordinary situation to get some statistics from

the system. Accordingly, she has other preferences regarding the decision context. Hence, the curation services composition module adapts itself and generates another scheme to meet Alice's needs. As for Bob, we assume that he is using the crisis management system to check the last recommendations to treat a new infectious disease. Thus, the system collects data from diverse sources like health institutions to generate these recommendations. We consider, in this example, the databases provided by health institutions. Accordingly, the curation service composition module initializes the MDP environment differently than in Alice's case by enabling curation services for batch and structured data curation, since Bob is more interested in the accuracy of the results. The curation services composition module adjusts Eq. 1 weights during the exploration/exploitation process to promote accuracy over response time. Hence, it generates a different curation service composition scheme that meets Bob's needs.

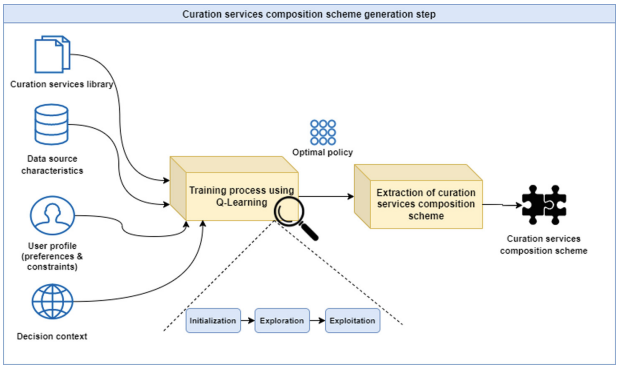


Fig. 3. Illustration of the curation services composition scheme generation step

5 Experiments and Results

In this section, we present the conducted experiments to assess the effectiveness and the performance of the curation service composition module. These experiments extend those that we presented in [14], which assess the reinforcement learning performance and adaptivity to changes. Specifically, in this paper, we evaluate the scalability of our approach regarding (1) the number of users, (2) the number of services, and (3) the adaptivity and the alignment with user requirements that we measure using cumulative rewards and evaluation scenarios. To do so, we applied our proposed method to generate curation services

composition scheme for an unstructured dataset[1], a semi-structured dataset[2], and a structured dataset[3]. Then, we compared the obtained results with the First-visit Monte Carlo, and Temporal-difference Learning [9], two well-known reinforcement learning algorithms, which we use to tackle the service composition problem. We adjusted the hyperparameters (e.g., learning rate) of each algorithm using several tests to identify the optimal configuration. Moreover, we assess the adaptivity of our approach regarding the changing data source characteristics, user preferences, and constraints using the cumulative reward.

Curation Services Composition Scalability. These experiments examine the scalability of our curation service composition method according to the number of simultaneous users and curation services. For this purpose, we relied on multithreading to simulate the curation services composition by several users. Thus, we developed threads, and each one simulates the curation service composition by one user. We defined the same input parameters (i.e., user preferences, constraints, etc.) for all the threads. Then, we executed and increased the number of threads progressively to examine the response time to these queries. Indeed, we considered the average execution time as a result. Figure 4a depicts the overall execution time according to the number of threads. Since we used different datasets to elaborate the experiments, we noticed that the data source type does not affect our approach performance. As for the scalability according to the number of services, we executed the curation services composition process for several services by increasing the Q-Table size. Indeed, as each entry of the Q-Table corresponds to one service, increasing the Q-Table size simulates the rising number of service instances. At each iteration, we executed the service composition three times and took the average execution time as a result. Consequently, the elaborated experiments showed that the service composition scheme is generated in near real-time using 12000 services. However, in our case, the Monte Carlo and Temporal-difference algorithms cannot generate a service composition scheme for more than 200 services. Following the experimental results, we noticed that our curation services composition method is scalable according to the number of users' queries and the curation services. Moreover, we can see that our proposed method outperforms the two reinforcement learning algorithms.

Cumulative Rewards. We conducted experiments on the abovementioned datasets to assess our curation service composition method's alignment with user expectations. As we adopted the reinforcement learning paradigm, we rely on the returned reward to evaluate the adaptivity of our proposed method to the QoS, user preferences, and constraints. Accordingly, we highlight that as the value of the reward increases, the composition method becomes more aligned with user needs. To do so, we designed a library of 17 existing services for structured

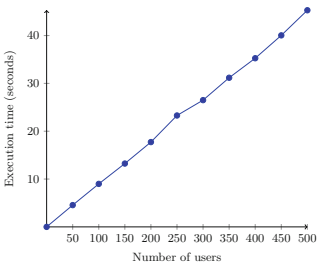
[1] https://archive.ics.uci.edu/ml/datasets/Health+News+in+Twitter: A dataset that contains health news from more than 15 major health news agencies such as BBC.

[2] https://www.ncbi.nlm.nih.gov/datasets/coronavirus/genomes/: A dataset provided by the National Center for Biotechnology Information (NCBI) that contains data about COVID-19 genomes.

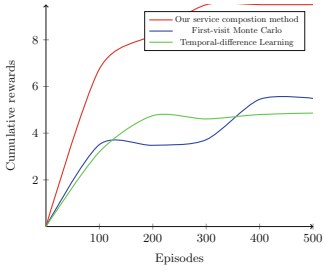
[3] https://github.com/LogIN-/fluprint: A structured dataset about Flu.

and unstructured/semi-structured data curation. Afterward, we defined similar experimental settings for all the tested algorithms. Specifically, we defined similar user preferences, QoS values, decision context, user constraints, and data source format. Considering these parameters, we executed our curation service composition method using the Q-Learning, the First-visit Monte Carlo, and the Temporal-difference learning algorithms to generate a curation services composition scheme for the three datasets. Then, we took the average of the rewards gained for the three datasets and presented them in Fig. 4b. As depicted in the Figure, our curation service composition aligns better with user needs than the First-visit Monte Carlo and Temporal-difference Learning algorithms, since it returns a higher cumulative reward. Indeed, the cumulative reward gained by our service composition method exceeds 9, while the maximum rewards returned by the other reinforcement learning algorithms are less than 6. We also investigated the adaptivity of our approach via the definition of evaluation scenarios. Through these scenarios, we generated curation services composition schemes for different data source types according to different user requirements. We examine the validity of the curation services composition scheme via the investigation of the convenience of its services according to the functional and non-functional requirements. To do so, we rely on the library of services described above that contains curation services for structured, semi-structured, and unstructured data sources. Specifically, we evaluated our approach's adaptivity according to (i) data source type, (ii) QoS and user preferences, and (iii) data source characteristics. Regarding the data source type, our approach has generated different services composition schemes for each data source type. For instance, the composition scheme for the semi-structured data source contains (Entity Extraction Service → Linking Service → Synonym Service), which is different from the one generated for the structured data source (Metadata Extraction Service → Descriptive Statistics Service → Missing Values Service → Terminology Extraction Service → Lexical Service → Rules Extraction Service → Entity Extraction Service → Linking Service → Synonym Service). Thus, the experiment results show that our approach generates convenient curation services schemes for each data source type (i.e., structured, semi-structured, unstructured). Regarding user preferences, we extended the library of services by defining three curation services for each curation task (e.g., PoS Tagging). Indeed, each curation service has different QoS values that are measured using the accuracy, availability, reliability, response time, reputation, and security quality dimensions. We treat in the first evaluation scenario a semi-structured data source, and we assume that the user expresses more interest in response time via the definition of the following preferences (Accuracy: 10%, Availability: 10%, Reliability: 10%, Response Time: 50%, Reputation: 10%, Security: 10%). Accordingly, our approach generated the following scheme (Stem Extraction Service → Synonym Extraction Service). Subsequently, we examined the services that constituted the scheme to check whether our service composition method has selected the most convenient services according to the user preferences. As we defined three curation services corresponding to each curation task, our method generates a composition scheme

that contains the least costly services in terms of response time. We consider the stem extraction services ST1, ST2, ST3, and the synonym extraction services SY1, SY2, SY3 which are candidates to constitute the abovementioned scheme. Hence, our method selected ST2 and SY1 which have the most Response Time QoS (i.e., ST1: 71%, ST2: 85%, ST3: 50%, SY1: 99%, SY2: 78%, SY3: 84%). Following this experiment, we changed the user preferences by promoting the accuracy dimension (Accuracy: 50%, Availability: 10%, Reliability: 10%, Response Time: 10%, Reputation: 10%, Security: 10%) to check whether this will have an impact on the curation services composition scheme. Hence, the generated composition scheme is constituted from ST3 and SY3 which are the most accurate services (i.e., ST1: 48%, ST2: 66%, ST3: 77%, SY1: 77%, SY2: 40%, SY3: 98%). Thus, based on the cumulative rewards values and the evaluation scenarios, we can deduce that our method aligns well with user requirements. We also investigated the alignment of our method with the treated data source characteristics. As we illustrated in Sect. 3, our framework extracts the characteristics of the data sources that are required for scheme generation. Following the characterization of a semi-structured data source containing URLs, the scheme generated by our method (Stem Extraction Service → URL Extraction Service → Entity Extraction Service → Synonym Extraction Service) contains a service dedicated to fetching data from URLs. We present another evaluation scenario in which we treat streaming data. By investigating the scheme generated for the streaming data (Streaming Concept Drift Service → Stem Extraction → Entity Extraction → Linking Service), we noticed that the generated scheme is different from the other schemes generated for batch data since it contains a service dedicated to streaming data. Hence, the results reveal our method ensures adaptivity since it aligns with the functional and non-functional requirements for data curation services scheme generation.



(a) Execution time per number of users



(b) Cumulative rewards by each algorithm

Fig. 4. Experiments results

Following the obtained results, we conclude that our method outperforms the Monte Carlo and Temporal-difference algorithms in terms of scalability, alignment with user needs, and execution time. We also noticed that our method has

the same performance for the different data source types. As a result, we believe it is appropriate for use in systems that handle heterogeneous data for different user needs in various decision contexts. Hence, it aligns with their needs to generate personalized outcomes. Moreover, it can be suitable in critical contexts since it is lightweight and fast. We highlight that the scope of our experiments covers only the service composition scheme generation and not services invocation.

6 Conclusion

We presented an original framework for batch and streaming data curation in data lakehouses. This paper sheds light on a novel adaptive curation services composition scheme generation method that constitutes the core of the presented framework. Hence, the originality of this work is the adaptivity throughout the entire process, starting from data collection to data curation passing by data quality evaluation, and data source characterization. The proposed method relies on the reinforcement learning algorithm to retrieve the convenient curation services composition according to the functional and non-functional requirements thanks to a reward function that considers the data source type, the decision context, the QoS values, and the users' preferences and constraints as well. We conducted several experiments to present the deterministic aspect of our proposal and illustrate its performance in comparison to well-known reinforcement learning algorithms, namely, First Visit Monte Carlo and Temporal difference algorithms. Experimental results show that our method outperforms the evaluated algorithms in terms of alignment with user expectations, the quality of the outcomes, and overall execution time. In future work, we plan to consider the trade-off between adaptive data curation services outcome and the accuracy of data analysis. The goal is to investigate data analysis scenarios within a data lakehouse to evaluate the effectiveness and accuracy of their curation.

References

1. Akoka, J., Comyn-Wattiau, I., Laoufi, N.: Research on big data - a systematic mapping study. Comput. Stan. Interfaces **54**, 105–115 (2017)
2. Beheshti, A., Vaghani, K., Benatallah, B., Tabebordbar, A.: CrowdCorrect: a curation pipeline for social data cleansing and curation. In: Mendling, J., Mouratidis, H. (eds.) CAiSE 2018. LNBIP, vol. 317, pp. 24–38. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92901-9_3
3. Bellomarini, L., et al.: Data science with Vadalog: knowledge graphs with machine learning and reasoning in practice. Futur. Gener. Comput. Syst. **129**, 407–422 (2022)
4. Konstantinou, N., et al.: VADA: an architecture for end user informed data preparation. J. Big Data **6**(1), 1–32 (2019). https://doi.org/10.1186/s40537-019-0237-9
5. Lauras, M., Truptil, S., Bénaben, F.: Towards a better management of complex emergencies through crisis management meta-modelling. Disasters **39**(4), 687–714 (2015)

6. Leonelli, S.: Classificatory theory in data-intensive science: the case of open biomedical ontologies. Int. Stud. Philos. Sci. **26**(1), 47–65 (2012)
7. Lord, P., Macdonald, A., Lyon, L., Giaretta, D.: From data deluge to data curation. In: In Proceedings of the 3th UK e-Science All Hands Meeting, pp. 371–375 (2004)
8. Maccioni, A., Torlone, R.: KAYAK: a framework for just-in-time data preparation in a data lake. In: Krogstie, J., Reijers, H.A. (eds.) CAiSE 2018. LNCS, vol. 10816, pp. 474–489. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_29
9. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. The MIT Press, Cambridge (2018)
10. Szepesvári, C.: Algorithms for reinforcement learning, vol. 9 (2010)
11. Tempini, N.: Data curation-research: practices of data standardization and exploration in a precision medicine database. New Genet. Soc. **40**, 73–94 (2020)
12. Wang, H., Zhou, X., Zhou, X., Liu, W., Li, W., Bouguettaya, A.: Adaptive service composition based on reinforcement learning. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6470, pp. 92–107. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17358-5_7
13. Weatherall, J., et al.: Clinical trials, real-world evidence, and digital medicine. In: The Era of Artificial Intelligence. Machine Learning, and Data Science in the Pharmaceutical Industry, pp. 191–215. Academic Press, Cambridge (2021)
14. Zouari, F., Ghedira, C., Kabachi, N., Boukadi, K.: Towards an adaptive curation services composition based on machine learning. In: IEEE International Conference on Web Services (ICWS), pp. 73–78 (2021)