Determining Query Readiness for Structured Data

Farid Alborzi[1(✉)], Rada Chirkova[1], Jon Doyle[1], and Yahya Fathi[2]

[1] Computer Science Department, North Carolina State University, Raleigh, NC, USA
{falborz,rychirko,jon_doyle}@ncsu.edu
[2] Industrial and Systems Engineering Department, North Carolina State University, Raleigh, NC, USA
fathi@ncsu.edu

Abstract. The outcomes and quality of organizational decisions depend on the characteristics of the data available for making the decisions and on the value of the data in the decision-making process. Toward enabling management of these aspects of data in analytics, we introduce and investigate *Data Readiness Level (DRL),* a quantitative measure of the value of a piece of data at a given point in a processing flow. Our DRL proposal is a multidimensional measure that takes into account the relevance, completeness, and utility of data with respect to a given analysis task. This study provides a formalization of DRL in a structured-data scenario, and illustrates how knowledge of rules and facts, both within and outside the given data, can be used to identify those transformations of the data that improve its DRL.

Keywords: Big data quality · Big data analytics and user interfaces · Data readiness level · Data quality measurement · Data quality improvement

1 Introduction

Organizations around the world increasingly apply analytics to their data to enable and facilitate recognition of events of interest, prediction of future events, and prescription of needed actions. These activities are made possible by the information and knowledge that analytics extract from the input data. By influencing the nature and quality of the extracted knowledge, the input data impact the overall decision-making process. A way to view this is that every datum exhibits inherent qualities that contribute in different ways to the value of the datum in a specific decision-making process. Low value qualities indicate that the datum is not reliable and may lead to inferior decision making.

As an example, consider the task of linking (a) the information collected about passing cars at a toll booth by a license-plate recognition system (LPR), with (b) the Department of Motor Vehicles (DMV) information about the owners of the cars. Imperfect LPR readings of license-plate information at the toll booth,

S. Madria and T. Hara (Eds.): DaWaK 2015, LNCS 9263, pp. 3–14, 2015.
DOI: 10.1007/978-3-319-22729-0_1

as well as missing car or driver information in the DMV database would be indications of "low-value" qualities of the data, which may lead to failures in performing the linkage task. In this paper we address detecting and correcting data-quality problems such as those in this "Toll Booth" example.

Our focus in this paper is on the problem of determining whether the inherent quality of the available data meets, or at least can be improved to meet, the data-quality expectations of the decision makers that have access to the data. We address this problem by introducing methods for quantifying and improving the quality of data. Our first contribution is in introducing and investigating *data readiness level (DRL),* a quantitative measure of the value of the quality of a piece of data at a given point in a processing flow. As such, the DRL represents a paradigm shift from the qualitative nature of traditional exploratory data analysis towards a rigorous metrics-based assessment of the quality of data in various states of readiness. The intuition for DRL is the distance between the information that can be extracted from (i.e., the "information content" [17] of) the given data for the given task, and the information content required by the task. Our second contribution is in introducing approaches for improving the value of data quality (i.e., DRL) of the data.

We use "relevance" and "completeness" dimensions to define DRL. In the above Toll Booth example, suppose 10 % of the driver-name information in the LPR data set is represented by null values (i.e., missing information). By the approach introduced in this paper, the "completeness" dimension of the corresponding DRL value of the data set would be 90 %. One potential DRL-improving solution proposed in this paper would replace these null values with the appropriate driver-name values from an external source. This would potentially increase the quality of these data values all the way to 100 % in that DRL dimension.

The foundational fact concerning data readiness is that data-readiness assessments rely on knowledge extracted from the data and, in turn, constitute knowledge about the data. We consider two major goals for DRL:

(1) Determine whether the given data have information of sufficient quality with respect to the given analysis task that is to be performed on the data.
(2) If this determination process returns a negative answer, identify ways to increase the information quality of the data with respect to the task.

Our Contributions: In this paper we address these two goals, by formalizing the data-quality measure DRL and approaches to improve its value. The study covers the core case where the data are relational, and tasks are carried out via relational (SQL) queries. Further, we assume that all the "content" problems in the data arise from missing (`NULL`) values. We refer to these assumptions as *the relational setting with null values.*

The rest of this paper is structured as follows. Section 2 defines the data-quality measure DRL for the relational setting. Section 3 introduces operators that improve the DRL value of the data for the tasks at hand. Section 4 presents a use case for the proposed approaches. Section 5 summarizes related work; Section 6 identifies som extensions of the relational setting.

Relation1 in database D1 at toll booth

row	plate	lane	CL	user	time	date
1	NMU45	3	1.00	NULL	09:20:16	07/04/2014
2	STA00	1	0.73	n1	09:20:03	07/04/2014
3	ABWD9	3	0.85	n2	09:19:53	07/04/2014
4	TRC19	4	1.00	n3	09:19:52	07/04/2014

Relation2 in D2 at NC DMV

LicNum	name	plate	state
11156	n2	ABWD9	VA
78922	n1	STA00	VA
58556	w1	NMU45	NC
82659	n3	TRC19	MD

Fig. 1. Table Relation1 in database D1, and table Relation2 in database D2. (CL stands for "confidence level," and LicNum for "driver license number.")

2 Formalizing Data Readiness Level

In this section we formalize the notion of data readiness level for the case where the data are structured using the relational data model, and tasks are carried out using relational (SQL) queries. Further, we assume that all the "content" problems in the data arise from missing (NULL) values. (We call this core case *the relational setting with null values,* and discuss extensions in Sect. 6.) This section provides the definitions and intuition, as well as illustrations via the Toll Booth example of Sect. 1.[1]

2.1 Toll Booth Example — Traffic Flow Identification

We begin by providing details on the Toll Booth example of Sect. 1; the example will be used to illustrate the concepts and approaches introduced in this paper. For the traffic-management department of a city, consider the *task* of obtaining the names of owners of those cars that are registered in North Carolina (NC) and that entered the city through a toll road on July 4, 2014. Suppose the toll road operates toll booths equipped with a license-plate recognition system (LPR). LPR identifies the number on a license plate with some possibility of error, with a confidence-level value that is either in the interval [0,1] or equals NULL.

Suppose the database D1 at a toll booth includes a table Relation1 obtained from the LPR system, and database D2 at the NC DMV has table Relation2, see Fig. 1. The task at hand can be expressed over each of D1 and D2 as:

```
(Q1 over D1): SELECT user FROM Relation1
              WHERE state='NC' AND date='07/04/2014' AND CL>=0.8
(Q2 over D2): SELECT name FROM Relation2
              WHERE state='NC' AND date='07/04/2014' AND CL>=0.8
```

Please note that neither Q1 nor Q2 is to be posed directly over the respective database. Instead, we envision a user interface where the data first get checked for their data readiness level with respect to each query. The query is posed over the database only after the data readiness level of the data has been improved

[1] Due to the page limit, the details can be found in the online version [2] of this paper.

in a satisfactory way. In particular, the query Q1 is originally formulated correctly *with respect to the task at hand* independently of the available table, even though Q1 mentions the state attribute that is absent from Relation1. Thus, we envision that the query Q1 will not be rejected outright when posed directly on database D1. Rather, the DRL value of D1 can be improved by importing the state attribute, perhaps from database D2. Once the data readiness level of D1 is improved this way, Q1 will be syntactically correct for execution over D1.

2.2 Data Readiness Level: Intuition and Preliminaries

Intuitively, data-readiness judgments concern *utility* of the given data for the given task. In general, utility might depend on numerous factors. In our relational setting, we consider a very simple form of utility, where the readiness level of a data set with respect to (w.r.t.) a task is a tuple of "relevance" and "completeness" dimensions. Here, *relevance* represents how close the structure of the data is to the task requirements, and *completeness* represents the fraction of useful (non-NULL) values among the data values available for addressing the task. We provide a formalization of this two-dimensional DRL measure in Sect. 2.5. The definition there uses the formalizations of relevance (Sect. 2.3) and completeness (Sect. 2.4) of a given data set w.r.t. a given task.

As an illustration, the relevance of the database D1 of Sect. 2.1 is not "100 % satisfactory" w.r.t. the query Q1, as the state column mentioned in Q1 is absent from Relation1. Similarly, the completeness of D1 is not "100 % adequate" w.r.t. Q1, because the user name is NULL in the first row of Relation1.

We now begin introducing the DRL formalism, by specifying the notion of task in our core relational setting. We consider tasks carried out by SQL SELECT–FROM–WHERE queries without extra clauses. For our purposes, it is sufficient to consider the "signature" of each task. The *signature* of a SQL task Q is a triple

$$T(Q) = [S(Q), F(Q), W(Q)], \tag{1}$$

in which $S(Q)$ is the set of attributes in the SELECT clause of Q, $F(Q)$ is the set of relations in the FROM clause of Q, and $W(Q)$ is the set of attributes in the WHERE clause of Q. For instance, the task Q1 in our running example has the signature $T(Q1) = [\{\texttt{user}\}, \{\texttt{Relation1}\}, \{\texttt{state}, \texttt{date}, \texttt{CL}\}]$, and the task Q2 has the signature $T(Q2) = [\{\texttt{name}\}, \{\texttt{Relation2}\}, \{\texttt{state}, \texttt{date}, \texttt{CL}\}]$.

In addition to the signature for each task, we also consider the attributes available for the task in the data: The *available attributes* of a query Q, written $A(Q)$, is the set of all the attributes in all the relations in $F(Q)$. For the queries Q1 and Q2 of Sect. 2.1 we have $A(Q1) = \{\texttt{row}, \texttt{plate}, \texttt{lane}, \texttt{CL}, \texttt{user}, \texttt{time}, \texttt{date}\}$, and $A(Q2) = \{\texttt{LicNum}, \texttt{name}, \texttt{plate}, \texttt{state}\}$.

Finally, we assume that each user who poses a SQL task on the available data can also specify, for each relational attribute mentioned in the task, the nonnegative *relevance utility* ρ and *completeness utility* κ of the attribute for the task. (In the baseline case where the user does not specify individual utilities of attributes, we assume that along each of the relevance and completeness

Table 1. Relevance (ρ) and completeness (κ) utility values for task `Q1` of Sect. 2.1.

a	`user`	`state`	`date`	`CL`
$\rho(a, \texttt{Q1})$	4	1	4	2
$\kappa(a, \texttt{Q1})$	1	2	4	3

dimensions of DRL, the utility of each attribute mentioned in the task is 1.) For instance, for our example of Sect. 2.1, possible utilities of the attributes mentioned in task `Q1` are listed in Table 1.

2.3 The Relevance Dimension of DRL

Given a relational data set, a SQL task, and the relevance utilities of the task attributes (see Sect. 2.2), we combine these relevance utilities to obtain a measure of the relevance of the data set to the task.

Definition 1. *The* DRL relevance R *of a database* D *for a task with signature* $T = [S, F, W]$ *and available attributes* A*, is defined as the ratio of the weighted relevance utility of those task attributes that are available in the database, to the weighted relevance utility of all the attributes in the task:*

$$R(D,T) = \begin{cases} \dfrac{\sum_{a \in S \cap A} \rho(a,T) + \sum_{a \in W \cap A} \rho(a,T)}{\sum_{a \in S} \rho(a,T) + \sum_{a \in W} \rho(a,T)} & if\ F\ is\ in\ D \\ 0 & otherwise. \end{cases} \tag{2}$$

That is, if all the relations in the `FROM` clause of the task are in the database, we use a weighted average as the relevance of the database to the task. Otherwise, the corresponding relevance value is zero.

Suppose that in our example of Sect. 2.1, the relevance (ρ) utilities of attributes for task `Q1` are as given in Table 1. Then the relevance of `D1` for `Q1` is:

$$\begin{aligned} R(\texttt{D1}, \texttt{Q1}) &= R(\texttt{D1}, [\{\texttt{user}\}, \{\texttt{Relation1}\}, \{\texttt{state}, \texttt{date}, \texttt{CL}\}]) \\ &= \frac{4+0+4+2}{4+1+4+2} = 0.91 \end{aligned}$$

2.4 The Completeness Dimension of DRL

In our definition of completeness of a database for a task, we use the following notion. For a relation r in database D, denote by $|r|$ the number of rows in r, and for an attribute a in r, denote by $|r(a \neq \texttt{NULL})|$ the number of rows in r in which a non-`NULL` value appears for a. Then the *completeness degree* of attribute a w.r.t. relation r, denoted by $\phi(a, r)$, is defined as the ratio $|r(a \neq \texttt{NULL})|/|r|$. (When either $a \notin r$ or $|r| = 0$, we define $\phi(a, r)$ to be 0.)

Now for a data set and a SQL task with completeness utilities of its attributes (see Sect. 2.2), we combine these utilities with the respective completeness degrees to obtain a measure of the completeness of the data set for the task.

Definition 2. *The* DRL completeness K *of a database* D *for a task with signature* $T = [S, F, W]$ *and available attributes* A*, is defined as the ratio of the weighted completeness utility of those task attributes that are available in the database to the weighted completeness utility of all the attributes in the task:*

$$K(D,T) = \begin{cases} \dfrac{\sum_{a \in S} \kappa(a,T)\phi(a,F) + \sum_{a \in W} \kappa(a,T)\phi(a,F)}{\sum_{a \in S \cap A} \kappa(a,T) + \sum_{a \in W \cap A} \kappa(a,T)} & if\ F\ in\ D \\ 0 & otherwise. \end{cases} \quad (3)$$

That is, the DRL completeness of a database w.r.t. a task is computed as the sum of the completeness utilities of the task attributes weighted by the respective completeness degrees, normalized by the total possible (i.e., ideal) completeness.

Suppose that in our example of Sect. 2.1, the completeness (κ) utilities of attributes for task `Q1` are as given in Table 1. We obtain that $\phi(\texttt{user}, \texttt{Relation1}) = \phi(\texttt{date}, \texttt{Relation1}) = \phi(\texttt{CL}, \texttt{Relation1}) = 1$, that $\phi(\texttt{state}, \texttt{Relation1}) = 0$, and that $\phi(\texttt{user}, \texttt{Relation1}) = 3/4$. Then the completeness of `D1` for `Q1` is

$$\begin{aligned} K(\texttt{D1}, \texttt{Q1}) &= K(\texttt{D1}, [\{\texttt{user}\}, \{\texttt{Relation1}\}, \{\texttt{state}, \texttt{date}, \texttt{CL}\}]) \\ &= \frac{1.(3/4) + 2.(0) + 4.(1) + 3.(1)}{1.(1) + 2.(0) + 4.(1) + 3.(1)} = 0.97 \end{aligned}$$

2.5 Putting It Together: Data Readiness Level Tuples

As discussed in Sect. 2.2, we define the data readiness level as a tuple of relevance and completeness dimensions. Here, relevance represents the distance between the structure of the data and of the task, and completeness represents the availability of non-null values in the task-relevant attributes:

Definition 3. *The* data readiness level *of database* D *with respect to task* T *is a tuple of the relevance and completeness values of* D *w.r.t.* T*:*

$$DRL(D,T) = [R(D,T), K(D,T)]. \quad (4)$$

In our running example of Sect. 2.1, the DRL for database `D1` for task `Q1` is $DRL(\texttt{D1}, \texttt{Q1}) = [0.91, 0.97]$. By similar calculations, for database `D2` and task `Q2` we obtain that $DRL(\texttt{D2}, \texttt{Q2}) = [0.45, 1]$. (The intuition for the relevance dimension of this DRL value for `D2` and `Q2` comes from the observation that the structure of `Relation2` in `D2` does not match well the attributes mentioned in `Q2`. At the same time, `Relation2` has no null values, which explains the "perfect score" of 1 in the completeness dimension of the DRL value for `D2` and `Q2`.)

3 Improving Readiness Level of Data for Task at Hand

In Sect. 2 we formalized the notion of data readiness level for the relational setting with null values. In this section, we discuss actions that can be taken in this context, to improve the DRL value of a data set for a given task. Our contribution is in providing a taxonomy and descriptions of meta-operators, which we refer to

as "data-readying operators." The distinction between our meta-operators and data-improvement approaches in the literature (e.g., data-cleaning approaches) is that we consider as explicit inputs to the data-improvement process not only the data, but also its DRL value w.r.t the *task* to be performed on the data, and a threshold on the desirable DRL value. As a result, to become applicable to the specific data set and task at hand, each meta-operator that we discuss is to be *instantiated* with both data-specific and task-specific parameter values, as well as potentially with knowledge that is (external to the data and task, and) available to the meta-operator for the data-improvement purpose.

3.1 Taxonomy of DRL-Improving Operators

In our relational setting with null values, we consider meta-operators that improve the DRL of the data by increasing the information content [17] of the data. The specific meaning here (same as in [17]) is that each operator "repairs" the `NULL` values in the data, by converting these values, to the extent possible, into non-`NULL` values that conform to the real world modeled by the data set.

We classify meta-operators aimed at improving the DRL of the data in this context by using three categories: null-value cleaning operators, resource-changing operators, and database integrating operators. Figure 2 summarizes these as part of a taxonomy of operators that increase the information content [17] of the data. Among these operator types, null-value cleaning operators increase the completeness dimension of the DRL, by looking for null values present in the data and using one of several possible techniques to repair the null values. Resource-changing operators aim to increase the relevance of the data for the task, by finding alternative resources (relations) with the task-appropriate structure. Finally, database integrating operators look for data sets that can be integrated with the given data to improve its relevance *and* completeness.

3.2 Illustration Using the Running Example

We now illustrate the use of null-values cleaning operators via our running example of Sect. 2.1. As discussed above, meta-operators in this class improve the value of the completeness dimension of the DRL of the given data set, by repairing (i.e., replacing by meaningful values) null values present in the data. Some other classes of our data-readying operators are illustrated in Sect. 4. For further details, please refer to the online version [2] of this paper.

Among the null-values cleaning operators that we consider in our taxonomy, one approach to repairing null values of an attribute in a relation consists of importing into the relation the appropriate non-null values of a matching attribute in another table. At the meta-operator level, this approach accepts as inputs the data set, the task information, and knowledge about which attributes in the available relations represent the same concept in the subject-matter ontology. An instantiation of this meta-operator within a particular DRL-improving process would accept specific values of all these inputs and would then proceed to replace all the `NULL` values in the attribute of interest by the non-`NULL` values taken from another relation. The non-`NULL` values used to repair the `NULL`s in

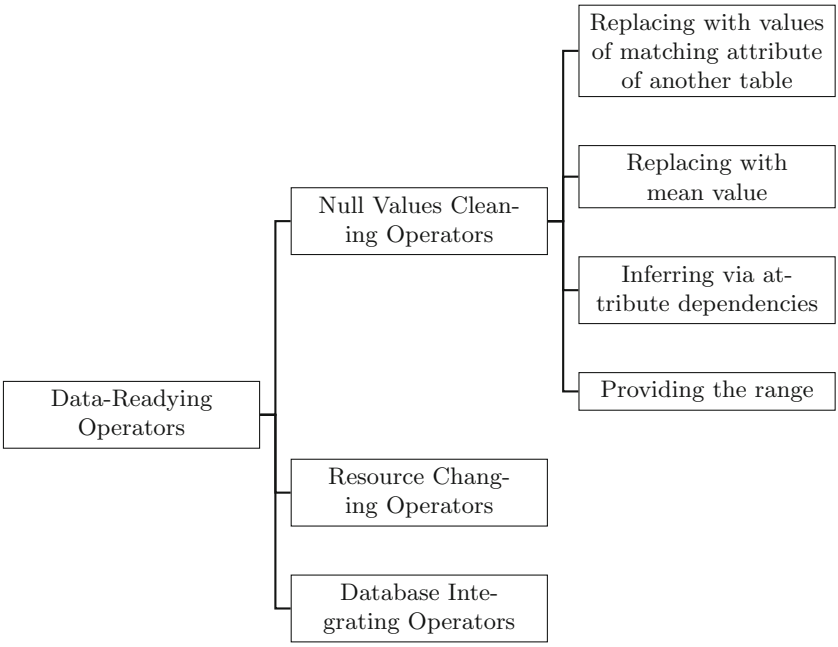

Fig. 2. Taxonomy of operators for increasing the information content of the data.

the attribute of interest would come from the same concept (i.e., attribute) and for the same real-world object (i.e., "appropriate" row in the relation) as those for the `NULL` value being replaced in the attribute of interest.

In our running example of Sect. 2.1, `Relation1` includes attributes `plate` and `user`. Suppose these attributes are linked to the concepts *PlateNumber* and *UserName*, respectively, in the available knowledge base. Further, suppose that attributes `plate` and `name` of `Relation2` are linked in the knowledge base to the same respective concepts. Then we say that `plate` in `Relation1` "concept-matches" `plate` in `Relation2`, and `user` in `Relation1` "concept-matches" `name` in `Relation2`. Observe that the `plate` attribute is an identifier in `Relation2`. Consequently, the instantiation of our meta-operator can replace the `NULL` values of the `user` attribute in `Relation1` by those (non-`NULL`) values of the `name` attribute of `Relation2` that correspond to the same value of the *PlateNumber* concept.

The output of applying this data-improvement operator to the `user` attribute of `Relation1` in the Toll Booth example is demonstrated in Table 2.

4 Use Case: Marketing via Targeted Mailings

To further illustrate the meta-operators described in Sect. 3, we introduce the following use case that complements our example of Sect. 2.1. In this use case, the marketing department of a company decides to contact all the customers

Table 2. Repaired NULL values of the user attribute in Relation1 of Sect. 2.1.

row	plate	lane	CL	user	time	date
1	NMU45	3	1.00	w1	09:20:16	07/01/2014
2	STA00	1	0.73	n1	09:20:03	07/01/2014
3	ABWD9	3	0.85	n2	09:19:53	07/01/2014
4	TRC19	4	1.00	n3	09:19:52	07/01/2014

who are over 20 years old and use the company's Plan B, to motivate them to switch to the company's Plan A. For this purpose, the company plans to use its database D3, with relations Sales and Customer as shown in Fig. 3.

Sales

id	custId	boughtPlan	date
1	103	Plan B	01/11/2015
2	102	Plan B	10/25/2014
3	104	Plan A	12/28/2014
4	102	Plan A	01/17/2014

Customer

id	name	address	currentPlan	age
101	David Smith	22nd St.	Plan A	NULL
102	Alfred Luck	20th St.	NULL	18
103	Daniel Bush	25th St.	Plan B	30
104	Goldy Elbetri	7th St.	NULL	NULL

Fig. 3. The Sales and Customer relations in database D3 for the Marketing use case.

The following task Q3, with signature [{name, address}, {Customer}, {age, currentPlan}], could be used to find the customers to be targeted in the mailing:

```
(Q3) SELECT name, address FROM Customer
     WHERE currentPlan = 'Plan B' AND age >= 20;
```

Since all the attributes mentioned in Q3 are in the data set D3, the relevance value of D3 for Q3 is 1. Assuming that the completeness utilities for Q3 are as in Table 3, the formulas in Sect. 2 yield $DRL(\texttt{D3}, \texttt{Q3}) = [1.0, 0.73]$.

Table 3. Completeness (κ) utilities for task Q3 in the Marketing use case.

a	name	address	currentPlan	age
$\kappa(a)$	1	4	4	2

Suppose the following facts, external to the data and the task, are known in this example: the values of the age attribute in the Customer relation are normally distributed; there are statistically enough observations with non-null values in that column of Customer; and most of the observations of customer age in Customer fall around the mean value. Then the null-value-cleaning operator

Table 4. Improved-quality `Customer` relation in the Marketing use case.

id	name	address	currentPlan	age
101	David Smith	22nd St	Plan A	24
102	Alfred Luck	20th St	Plan B	18
103	Daniel Bush	25th St	Plan B	30
104	Goldy Elbetri	7th St	Plan A	24

of Fig. 2 that replaces `NULL` values with the mean value of `24` can be applied to `age` in `Customer`. The results of improving the information content of the `age` attribute of the `Customer` relation are shown in Table 4.

Further, suppose that additional available knowledge prescribes replacing the `NULL` values of the attribute `currentPlan` in the `Customer` relation with the latest plan bought by the customer. The corresponding meta-operator, suitably instantiated, would look for tuples in the `Sales` relation that have the same value of `custId` as the value of `id` in the `Customer` relation. The meta-operator would then replace the associated `NULL` values of `currentPlan` in `Customer` with the information on the most recent transaction for the customer, which is represented by the tuple with the maximum value of `date` attribute. All this replacement knowledge can be encoded in a rule that is specific to the data set and task, but would be used by an instantiation of the generic meta-operator. The result of this meta-operator application to the `currentPlan` attribute in the `Customer` relation is shown in Table 4.

5 Related Work

Projects focused on defining and classifying data-quality dimensions have related the data format to syntactic criteria [9] and investigated the semantics of data values [13,14]. In measuring data quality, two directions have emerged: quality of conformance and quality of design. Quality of conformance aims to align an information system's existing data values and its design specifications, whereas in quality of design, checking the closeness of system specifications to the customer requirements is of interest [12]. Subjective data-quality assessments, such as those in quality of design, can be approached by distributing questionnaires among stakeholders [18]. In contrast, studies focused on objective measurements, such as quality of conformance, introduce and investigate descriptive metrics. The data-quality metrics introduced in the literature either assume fixed tasks in quantifying data quality, or do not consider tasks at all. In contrast, in this work we provide a framework for computing metrics of quality of the given data with respect to specific (but not fixed) data-processing tasks.

We formalize the notion of data relevance in a way similar to [15], where relevance is defined as "level of consistency between the data content and the area of interest of the user." While relevance has been addressed in several studies (e.g., [22,23]) as a data-quality dimension, to the best of our knowledge

no metric has been provided to quantify it in the literature. The framework proposed in this current work includes a metric for computing the current level of data relevance and for measuring the quality of relevance-improving solutions.

Completeness is defined in [18] as a fraction of non-null values of an attribute. A number of approaches have been proposed in data management for addressing problems caused by null values in the data, e.g., [4,5,7,17]. Directions of studies of incomplete information [17] in the literature include approaches based on representation systems and certain answers [1,11]), logical theory [19,20], and programming semantics [3,16]. Unlike previous work, in this study we introduce a formalization where completeness of the data can be improved based on knowledge that is external to the data and task at hand.

Knowledge bases have been used in a variety of applications [8]. Notably, [6] introduced a platform that employs user-imposed rules to repair data, based on violations of the available data-cleaning rules. At the same time, [6] does not focus on evaluating quality of the data in presence of specific data-processing tasks. In this study we introduce a framework that measures the quality value of data in a task-specific way, and uses external knowledge to improve the quality value of the data for the task at hand.

6 Conclusion and Future Work

In this paper we studied the problem of quantifying the readiness of a relational data set to handle tasks carried out via SQL queries, in presence of missing information in the form of null values. We formalized this data-readiness problem and proposed approaches for evaluating the relevance and completeness aspects of our data readiness measure in presence of a given data-oriented task. In addition, we developed a taxonomy of "data-readying" meta-operators that improve the information content of the data for the task at hand, in presence of knowledge available about the data or the task. The proposed formalization can be extended to quantifying the data-quality dimensions (e.g., relevance and completeness) of a data set with respect to a collection of typical tasks posed to the system.

This initial study of the core case of data readiness involves several simplifying assumptions about the structure of data sets, tasks, and task-dependent utility of the data. Our future work aims to relax these assumptions, e.g., to extend the framework to allow subqueries in the tasks; to consider types of data-quality issues beyond null values; to enable treatment of more complex and realistic data-analysis scenarios and data-quality dimensions; and to consider data models beyond the relational model. Other extensions involve developing techniques to capture and express utility information and probabilistic properties of data-readying operators.

Acknowledgment. This material is based upon work supported in whole or in part with funding from the Laboratory for Analytic Sciences (LAS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, San Diego (1995)
2. Alborzi, F., Chirkova, R., Doyle, J., Fathi, Y: Determining query readiness for structured data. Technical Report (which is not a publication) TR-2015-6, NCSU, 2015. http://www.csc.ncsu.edu/research/tech/reports.php
3. Buneman, P., Jung, A., Ohori, A.: Using power domains to generalize relational databases. TCS **91**(1), 23–55 (1991)
4. Codd, E.F.: Extending the database relational model to capture more meaning. ACM TODS **4**(4), 397–434 (1979)
5. Codd, E.F.: Understanding relations (installment #7). FDT - Bull. ACM SIGMOD **7**(3), 23–28 (1975)
6. Dallachiesa, M., Ebaid, A., Eldawy, A., Elmagarmid, A., Ilyas, I. F., Ouzzani, M., Tang, N.: NADEEF: a commodity data cleaning system. In: ACM SIGMOD, pp. 541–552 (2013)
7. Date, C.J.: Database in Depth - Relational Theory for Practitioners. OReilly, Sebastopol (2005)
8. Deshpande, O., Lamba, D.S., Tourn, M., Das, S., Subramaniam, S., Rajaraman, A., Doan, A.: Building, maintaining, and using knowledge bases: a report from the trenches. In: ACM SIGMOD, pp. 1209–1220 (2013)
9. Eppler, M.J.: Managing Information Quality: Increasing the Value of Information in Knowledge-intensive Products and Processes. Springer, Berlin (2006)
10. Gardyn, E.: A Data Quality Handbook for a Data Warehouse. Infrastructure. IQ, 267–290 (1997)
11. Grahne, G.: The Problem of Incomplete Information in Relational Databases. Springer, Berlin (1991)
12. Heinrich, B., Helfert, M.: Analyzing data quality investments in CRM: a model-based approach. In: Eighth International Conference on Information Quality, pp. 80–95 (2003)
13. Heinrich, B., Klier, M., Kaiser, M.: A procedure to develop metrics for currency and its application in CRM. J. Data Inf. Qual. **1**(1), 5 (2009)
14. Hinrichs, H.: Datenqualitatsmanagement in Data Warehouse-Systemen. Ph.D. thesis, Universitat Oldenburg (2002)
15. Kulikowski, J.L.: Data quality assessment. In: Ferraggine, V.E., Doorn, J.H., Rivero, L.C. (eds.) Handbook of Research on Innovations in Database Technologies and Applications, 378–384. Hershey, PA (2009)
16. Libkin, L.: A semantics-based approach to design of query languages for partial informatio. In: Thalheim, B., Libkin, L. (eds.) Semantics in Databases. LNCS, vol. 1358, pp. 170–208. Springer, Berlin (1995)
17. Libkin, L.: Incomplete data: what went wrong, and how to fix it. In: PODS, 1–13. ACM (2014)
18. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Comm. ACM **45**(4), 211–218 (2002)
19. Reiter, R.: On closed world data bases. Logic Data Bases **33**, 55–76 (1977)
20. Reiter, R.: Towards a logical reconstruction of relational database theory. Conceptual Model. **33**, 191–233 (1982)
21. Teboul, J.: Managing Quality Dynamics. Prentice Hall, New York (1991)
22. Wand, Y., Wang, R.Y.: Anchoring data quality dimensions in ontological foundations. Commun. ACM **39**(11), 86–95 (1996)
23. Wang, R.Y., Strong, D.M.: Beyond accuracy: what data quality means to data consumers. J. Manage. Inf. Syst. **12**(4), 5–34 (1996)