Measuring and Comparing Effectiveness of Data Quality Techniques

Lei Jiang[1], Daniele Barone[2], Alex Borgida[1,3], and John Mylopoulos[1,4]

[1] Dept. of Computer Science, University of Toronto
[2] Dept. of Computer Science, Università di Milano Bicocca
[3] Dept. of Computer Science, Rutgers University
[4] Dept. of Information Engineering and Computer Science, University of Trento

Abstract. Poor quality data may be detected and corrected by performing various quality assurance activities that rely on techniques with different efficacy and cost. In this paper, we propose a quantitative approach for measuring and comparing the effectiveness of these data quality (DQ) techniques. Our definitions of effectiveness are inspired by measures proposed in Information Retrieval. We show how the effectiveness of a DQ technique can be mathematically estimated in general cases, using formal techniques that are based on probabilistic assumptions. We then show how the resulting effectiveness formulas can be used to evaluate, compare and make choices involving DQ techniques.

Keywords: data quality technique, data quality measure, data quality assurance.

1 Introduction

The poor quality of data constitutes a major concern world-wide, and an obstacle to data integration efforts. Data of low quality may be detected and corrected by performing various quality assurance activities that rely on techniques with different efficacy and cost under different circumstances. In some cases, these activities require additional data, changes in the database schema, or even changes in core business activities. For example, consider the relation schema *Person(sin, name, address)*, which intends to record a person's social insurance number, name and address. Due to the decision to represent an address value as single string, no obvious integrity constraints or other automatically enforceable techniques can be specified on the components of the address value [1]. In particular, one cannot detect missing street, city, etc. using "not null" constraints, because nothing is said in the schema about the exact format of address values.

In [2], we proposed a goal-oriented database design process, and extended in [3] to handle data quality goals. The quality design process starts with a conceptual schema, which is then augmented by a set of high level *data quality goals* (e.g., "accurate student data"). These goals are gradually decomposed into concrete *data quality problems* to be avoided (e.g., no "misspelled student names"). For each such problem, a list of *risk factors* (i.e., potential causes) and *mitigation plans* (i.e., potential solutions) is presented. The main component of a mitigation plan is a *design proposal* consisting of a revised original schema and a set of data quality (DQ) techniques it supports.

P. van Eck, J. Gordijn, and R. Wieringa (Eds.): CAiSE 2009, LNCS 5565, pp. 171–185, 2009.

In this paper, we take the next step of proposing a quantitative approach for measuring and comparing the effectiveness of DQ techniques used in quality assurance activities. The main contributions of this paper include: (i) the definitions of effectiveness measures for DQ techniques, based on the well-established notions of precision, recall and F-measure; (ii) formal techniques for estimating the expected effectiveness scores for a technique (on a wider range of possible instances of a database), based on probabilistic assumptions about the occurrence of errors in data values and confounding factors; these techniques result in effectiveness formulas parametrized by variables introduced by these assumptions; (iii) analysis and comparison of DQ techniques and their respective strengths in terms of the subranges of values of the parameters in the effectiveness formulas.

The rest of the paper is organized as follows. We first discuss briefly the main concepts in our approach in Section 2. We then present the definitions of our effectiveness measures in Section 3, and show examples of calculating effectiveness scores when a database instance is available. Next, a general pattern is identified for the formal estimation of the expected effectiveness scores based on the probabilistic assumptions, and is applied to several DQ techniques in Section 4.1. The resulting effectiveness formulas provide input for the what-if analysis in Section 4.2, in which we evaluate a single DQ technique and compare multiple ones under different scenarios. Finally, we review the related work in Section 5, and conclude and point out to our future work in Section 6.

2 Main Concepts

2.1 DQ Techniques

The core concept in our approach is a *DQ technique*, which is, broadly speaking, any automatic technique that can be applied to data in a quality assurance activity, in order to assess, improve and monitor its quality. This includes techniques to standardize data of different formats, to match and integrate data from multiple sources, and to locate and correct errors in data values[1]. In this paper, we focus on DQ techniques that automatically enforce a rule of the form "if *condition* then *action*", where *condition* checks violation of some integrity constraint, and *action* produces either deterministic or probabilistic decisions regarding quality of data being examined (e.g., to mark values as possibly erroneous, to suggest possible corrections to erroneous values). Following example illustrates two simple rules.

Example 1. Consider the relation schema *Person* again. Suppose we are especially concerned with the quality of name values. In this case, we can modify this schema by adding a second name attribute as in *Person′(sin, name, address, name′)*, with the intention of modifying the workflow so that names are entered twice (by the same or different persons), and then detect errors by comparing the two name entries. The revised schema makes it possible to specify and enforce following rule, "for each tuple t inserted in *Person′*, if $t.name \neq t.name'$ then mark the tuple $\langle t.name, t.name' \rangle$ as erroneous". Another way to provide quality assurance for names, without changing the schema of *Person*, is to keep a table of valid names, L_{name} . This allows to specify and enforce the rule, "if $\neg(t.name \in L_{name})$, then mark t.name as erroneous". □

2.2 Effectiveness of DQ Techniques

Different DQ techniques may have different efficacy and cost under different circumstances. The effectiveness of a DQ technique is determined both by the nature of the technique and the particular values and errors in the data being examined. Following example explains the concept of effectiveness in the context of DQ techniques.

Example 2. Consider a simple conditional functional dependency ϕ = *[country-code = 44, area-code = 131]* $\rightarrow$ *[city-name = Edinburgh]* [4,5][1]. If it is the case that city name has much higher possibilities of having errors than country code and area code, violation of ϕ is more likely an indication of erroneous city name values than others. In this case, a DQ technique may check for violation of ϕ, and mark the city name value in a tuple as possibly erroneous whenever the tuple violates ϕ. A set of tuples marked by this DQ technique then needs to be presented to a domain expert who will make the final decision. To minimize human effort, ideally a city name value is actually erroneous if and only if the tuple containing this value is in the returned set. However, this is unlikely to be true due to (comparably small amount of) errors in country code and area code values. Effectiveness of ϕ measures its ability to produce "good" sets of tuples compared to the ideal set, for a given database instance or a range of instances. □

2.3 Effectiveness Measures, Scores and Formulas

With respect to a particular database instance, an *effectiveness score* is assigned to a DQ technique. To obtain such effectiveness scores, the first step is to adopt a set of *effectiveness measures*, such as precision, recall and F-measure from Information Retrieval. Then, the DQ technique is applied to a database instance (for which quality of data is already known, e.g., through manual assessment); and the effectiveness scores of the DQ technique is calculated by comparing its output with existing knowledge of the instance. There are several limitations for this approach. First, a database instance may not always be available (e.g., when designing a new schema) or only partially available (e.g., when modifying an existing schema). Second, the effectiveness scores only tell us how the DQ technique performs on one snapshot of the database.

Therefore, it is often necessary to consider how a DQ technique performs on average over a range of possible instances of the database. This leads to the *expected effectiveness score* of a DQ technique. To obtain expected effectiveness score of a DQ technique, one can first derive an *effectiveness formula* of the DQ technique. This requires making probabilistic assumptions about the occurrence of errors in data values and confounding factors. By *confounding factors*, we mean special events that may "confuse" a DQ technique, making it less effective. For example, the country name "Australia" may be misspelled as "Austria", which is still a valid country name; such an error cannot be detected using a technique based on a country name lookup table. The resulting effectiveness formulas, can be evaluated and compared by fixing some parameters in formulas and allowing the others to vary.

[1] A conditional functional dependency is formally defined as a pair of a regular functional dependency and a pattern tableau. Here we are using an abbreviated notation.

3 Effectiveness Measures

In this section, we present the definitions for our effectiveness measures. We first explain the general idea and give the basic definitions for these measures, and then extend them to accommodate errors of different types and data values from multiple attributes.

3.1 Basic Definition

Effectiveness represents the ability to produce a desired result (in order to accomplish a purpose). For DQ techniques, the purpose is to assess, improve, etc. quality of data. This gives rise to measures for assessability, improvability, etc. In what follows, we concentrate on assessability measures; we defer a detailed treatment of other types of effectiveness measures to a later report.

Assessability represents an DQ technique's ability to effectively detect erroneous data values. Normally, this ability can only be measured if we have access to the reality (i.e., is an erroneous value marked by a DQ technique really an error). When not accessible, we need an approximation of the reality, possibly obtained through some manual quality assurance activity. The precise meaning of "assessability" depends on what do we mean by "erroneous" and "data value". To begin with, we assume that each DQ technique assesses quality of data in a single attribute, and classifies the attribute values into two categories: those with some error, and those without. In Section 3.2, we relax these limitations.

Inspired by Information Retrieval, we define assessability measures in terms of precision, recall and F-measure [6]. More specifically, let S be a relation schema, A be an attribute in S, and I be an instance of S. Consider a DQ technique T, which is applied to I in order to assess quality of A values. Equation 1 and 2 defines the precision and recall [6] for T with respect to I and A; these two measures are combined in Equation 3 into F-measure [6], where β is a constant that represents the importance attached to recall relative to precision.

$$precision(T, I, A) = \frac{TP(T, I, A)}{TP(T, I, A) + FP(T, I, A)} \tag{1}$$

$$recall(T, I, A) = \frac{TP(T, I, A)}{TP(T, I, A) + FN(T, I, A)} \tag{2}$$

$$F_\beta(T, I, A) = \frac{(1 + \beta^2) \times precision(T, I, A) \times recall(T, I, A)}{\beta^2 \times precision(T, I, A) + recall(T, I, A)} \tag{3}$$

The values TP, FP and FN represent the number of true positives, false positives and false negatives respectively, and are explained more clearly below. Example 3 shows how the assessability scores can be calculated using the sampling approach.

- *TP(T, I, A)* = the number of erroneous A values in I, correctly marked by T as being erroneous
- *FP(T, I, A)* = the number of non-erroneous A values in I, incorrectly marked by T as being erroneous

– *FN(T, I, A)* = the number of erroneous A values in I, not marked by T as being erroneous, but should have been

Example 3. Consider the *Person* schema again. Suppose from one of its instances, I_{Person}, 10 tuples are selected as the sample. After performing some manual quality assurance activity on the sample, 3 erroneous name values are identified and the correct values are obtained. Table 1(a) shows the result of this manual activity, where the name value is erroneous iff *err* = "1"; $name^{new}$ is used to record the suggested name values[2].

Now consider a DQ design proposal P_1(*Person′(sin, name, address, name′)*, T_{equal}), in which the *Person* schema is revised to *Person′*, and T_{equal} is a DQ technique that enforces the rule: "for each tuple t inserted in an instance of *Person′*, if *t.name* $\neq$ *t.name′* then mark *t.name* as erroneous." An instance $I_{Person'}$ of *Person′* is generated by starting with data from I_{Person} and obtaining independent values for the new attribute *name′*.

Suppose we need to know how effective T_{equal} is in assessing *name* values in $I_{Person'}$. First, we select the same 10 tuples from $I_{Person'}$ as the sample, and obtain the quality assessments on the sample using T_{equal}, as shown in column err of Table 1(b). By comparing Table 1(b) with Table 1(a), we obtain following numbers TP = 2 (due to Tuple 006 and 009), FP = 2 (due to Tuple 001 and 008), and FN = 1 (due to Tuple 004). The assessability scores for T_{equal} on this sample (when $\beta = 1$) are: $precision(T_{equal}, I_{Person'}, name)$ = 0.5, $recall(T_{equal}, I_{Person'}, name)$ = 0.67, and $F_1(T_{equal}, I_{Person'}, name)$ = 0.57. □

Table 1. Calculation of effectiveness scores using the sampling approach

(a) Quality of *name* values in I_{Person}

sin	name	err	$name^{new}$
001	Kelvin	0	
002	Michelle	0	
003	Jackson	0	
004	Alexander	1	Alexandre
005	Maria	0	
006	Tania	1	Tanya
007	Andrew	0	
008	Christopher	0	
009	Michale	1	Michael
010	Matthew	0	

(b) DQ annotation for *name* values in $I_{Person'}$ using T_{equal}

sin	name	name′	err
001	Kelvin	Kelvn	1
002	Michelle	Michelle	0
003	Jackson	Jackson	0
004	Alexander	Alexander	0
005	Maria	Maria	0
006	Tania	Tanya	1
007	Andrew	Andrew	0
008	Christopher	Christophor	1
009	Michale	Michael	1
010	Matthew	Matthew	0

(c) DQ annotation for *name* values in $I_{Person'}$ using $T_{equal-prob}$

sin	name	name′	err	err′
001	Kelvin	Kelvn	0.5	0.5
002	Michelle	Michelle	0	0
003	Jackson	Jackson	0	0
004	Alexander	Alexander	0	0
005	Maria	Maria	0	0
006	Tania	Tanya	0.5	0.5
007	Andrew	Andrew	0	0
008	Christopher	Christophor	0.5	0.5
009	Michale	Michael	0.5	0.5
010	Matthew	Matthew	0	0

3.2 Extensions

We may be interested in measuring the effectiveness of a DQ technique with respect to particular types of errors, instead of considering all possible ones. For example, a lookup-table based DQ technique is very effective in detecting syntactic but not semantic accuracy errors [1]. In this case, the assessability scores can be calculated using Equation 1 and 2 in the same way as before, except that we only consider errors of the specified types when counting TP, FP and FN.

[2] The $address$ values are omitted here and thereafter.

Equation 1 and 2 work for DQ techniques whose output involve a single attribute. In some case, the result of a DQ technique may involve values of a set $X = \{A_1, \ldots, A_n\}$ of attributes. There are two ways to look at this situation, which lead to two different solutions. In one view, we may treat a tuple $t.X$ as a single value (i.e., $t.X$ is erroneous if any of $t.A_1, \ldots, t.A_n$ is). Then we can calculate assessability scores of a DQ technique using modified versions of Equation 1 and 2, where $precision(T, I, A)$ and $recall(T, I, A)$ are replaced with $precision(T, I, X)$ and $recall(T, I, X)$ respectively. In another view, we introduce the notion of uncertainty. This leads to a more general solution. When a DQ technique marks a tuple $t.X$ as being erroneous, it essentially marks each individual value $t.A_1, \ldots, t.A_n$ in the tuple as being erroneous with certain probability. If those probabilities can be estimated, we can still treat each attribute individually, but allow the assessment result to be a number between 0 and 1. Example 4 illustrates the second view.

Example 4. Let us consider another DQ design proposal P_2(*Person′(sin, name, address, name′)*, $T_{equal-prob}$), where $T_{equal-prob}$ is same as T_{equal} in P_1, except that it marks the whole tuple *t[name, name′]* as being erroneous when $t.name \neq t.name'$. Following the second view, if we assume that a *name* and *name′* value have the same probability of being wrong, Table 1(c) shows the output of $T_{equal-prob}$ applied to the same sample of $I_{Person'}$ (as in Example 3). Notice, *err* = "0.5" (respectively *err′* = "0.5") means the *name* (respectively *name′*) value is marked as erroneous with the 0.5 probability.

In this case, a real erroneous *name* value, being marked as erroneous with 0.5 probability (e.g., Tuple 006), counts for 0.5 toward *TP* and 0.5 toward *FN*. By comparing this table with Table 1(a), we can obtain following numbers: TP = 1 (due to Tuple 006 and 009), FP = 1 (due to Tuple 001 and 008) and FN = 2 (due to Tuple 004, 006 and 009). The assessability scores for $T_{equal-prob}$ on this sample can then be calculated using this numbers. □

4 Estimating and Comparing Expected Effectiveness Scores

4.1 Formal Approach

The above examples show the calculation of assessability scores for a DQ technique on a particular database instance. In this section, we show how assessability scores can be estimated without applying the DQ technique to data. More specifically, we show how to obtain the expected assessability scores for a DQ technique based on probabilistic assumptions. This approach can be divided into four steps: (1) setting the stage, (2) making probabilistic assumptions, (3) calculating probabilities for the events of interests, and (4) formulating assessability scores. In what follows, we illustrated this approach on several DQ techniques.

Introducing Redundancy. Although duplicating an attribute as we have shown in previous examples may seem simplistic, the idea of using redundancy checks (e.g., checksum) to protect the integrity of data has long been practiced in computer communication, and also been proposed for detecting corruption in stored data [7]. More generally, partial redundancy is the basis of many integrity constraints (e.g., correlations between phone area codes and postal codes).

Step 1: setting the stage. In general, given a relation schema S, we are interested in DQ design proposals of the form $P_{redundancy}(S', T_{B \neq f(X)})$, where S' contains all attributes in S plus a new attribute B, and $T_{B \neq f(X)}$ enforces the rule "for each tuple t inserted in an instance of S', if $t.B \neq f(t.X)$ then mark $t.X$ as erroneous"; here X is a subset of attributes in S, and f represents some computable function. For example, X may contain a single attribute *birthdate* and B is the attribute *age*; f computes the current age from the date of birth[3]. In what follows, we illustrate the formal approach for the case where X contains a single attribute A and f is the identity function, i.e., for the DQ technique $T_{B \neq A}$. More general cases can be handled in a similar way.

Step 2: making probabilistic assumptions. The main factor that affects the assessability scores for $T_{B \neq A}$ is the occurrence of errors in the attributes A and B. For the rest of the paper, we make several independence assumptions about values and errors in general: (i) the probability that a value will be wrong is independent of the value itself, and (ii) the probability of an error occurring in one attribute is independent of those of the other attributes.

To simplify the analysis here, we will assume that the probability of a A value or B value being incorrect is the same — denoted by p. If we use $Err^{t.A}$ to name the event that the recorded value in $t.A$ does not correspond to the real one and use $Cor^{t.A}$ to mean the converse, this assumption can be stated symbolically as pr($Err^{t.A}$) = pr($Err^{t.B}$) = p, where pr(E) represents the probability of an event E. Before we proceed further, we need to recognize that there is the possibility that both $t.A$ and $t.B$ are incorrect yet contain the same erroneous value; in this case, these errors "cancel out" as far as the DQ technique $T_{B \neq A}$ is concerned (since they cannot be detected by $T_{B \neq A}$). We call this situation "error masking", which is a particular type of confounding factors. Let us say that such masking will happen only with probability $1 - c_1$.

Step 3: calculating probabilities for the events of interests. To estimate the assessability scores, we are interested in events concerning a tuple t (i) whether $t.A$ has an error, and (ii) whether a DQ problem is signaled by $T_{B \neq A}$. This estimation has to be adjusted for error masking. To compute the expected values for *TP*, *FP* and *FN*, we will actually compute the probabilities of events concerning a particular tuple t, and then multiply this by the number of tuples in the relation.

First, true positives occur when $t.A$ has an error (probability p) that is correctly signaled by $T_{B \neq A}$. This happens when either $t.B$ is correct (prob. $(1 - p)$) or $t.B$ is incorrect (prob. p) but different from $t.A$ (prob. c_1); this yields probability: pr($Err^{t.A} \bigwedge Cor^{t.B}$) + pr($Err^{t.A} \bigwedge Err^{t.B} \bigwedge (t.A \neq t.B)$) = $p \times (1 - p) + p \times p \times c_1$.

False negatives occur when $t.A$ has an error that is not signaled by $T_{B \neq A}$, because error masking occurs (which requires $t.B$ to contain the exact same error); this has probability: pr($Err^{t.A} \bigwedge Err^{t.B} \bigwedge (t.A = t.B)$) = $p \times p \times (1 - c_1)$.

False positives occur when $t.A$ has no error yet $T_{B \neq A}$ signals a problem, which arises according to our rule when $t.B \neq t.A$ (i.e., when $t.B$ has an error); this has probability: pr($t.A^{cor} \bigwedge t.B^{err}$) = $(1 - p) \times p$.

[3] A variant of $T_{B \neq f(X)}$ replaces the condition "$t.B \neq f(t.X)$" with "$d(t.B, f(t.X)) > \delta$"; so instead of requiring $t.B$ and $f(t.X)$ to be exactly the same, it only requires their distance (measured by d) be less than a constant δ.

Step 4: formulating assessability scores. Given the probabilities obtained in Step 3, the expected number of true positives, false positives and false negatives can be calculated as the number of tuples (say N) times the respective probability as following: $TP(T_{B\neq A}, A) = N\times(p(1-p)+p^2c_1)$; $FN(T_{B\neq A}, A) = N\times p^2(1-c_1)$; $FP(T_{B\neq A}, A) = N \times (1-p)p$; The expected assessability scores for $T_{B\neq A}$ can then be obtained by plugging these numbers into Equation 1, 2 and 3. Since N appears both in the numerator and denominator, it will cancel out, resulting in the effectiveness formulas in Table 2 (Section 4.2).

Using Lookup Tables. For an attribute with a standardized (and finite) domain, such as country name or postal code, a common DQ technique is to check its values against a lookup table for the attribute. Attributes with enumerated value domains (such as *gender*) also offer this possibility.

Step 1: setting the stage. Given the original schema S and an attribute A in S, we are interested in DQ design proposals of the form $P_{lookup}(S, T_{L_A})$, where T_{L_A} is the DQ technique that detects errors in A values using a lookup table L_A. In what follows, we illustrate the formal approach for this type of DQ techniques.

Step 2: making probabilistic assumptions. We make two passes through this analysis, in order to account for two different sources of problems. First, we assume as before there is a probability p that the recorded value of $t.A$ is incorrect. In this case, error masking occurs when this erroneous value is still a valid value in the domain of A (e.g., "Australia" vs "Austria") – an event to which we assign probability c_2. If we use $Valid^{t.A}$ to name the event that the value $t.A$ is valid and $Invalid^{t.A}$ to mean the converse, we can represent these assumptions using following conditional probabilities: pr($Valid^{t.A}|Err^{t.A}$) = c_2 and pr($Invalid^{t.A}|Err^{t.A}$) = $1-c_2$.

Second, we consider the possibility of the lookup table being imperfect, which is another type of confounding factors. In particular, we allow a probability s that some value (e.g., the name of a newly independent country) is missing from the lookup table L_A[4]. If we use $L_A^{t.A}$ to name the event that the value $t.A$ is contained in L_A, and $L_A^{\neg t.A}$ to mean the converse, we have pr($L_A^{t.A}$) = $1-s$ and pr($L_A^{\neg t.A}$) = s. Notice here we are implicitly assuming that pr($L_A^{t.A}$) is independent from the characteristics of $t.A$ values (e.g., $name$ values of different length or in different languages).

Step 3: calculating probabilities for the events of interests. In the first case (i.e., assuming a perfect lookup table), true positives occur when $t.A$ is incorrect and the value is not in the lookup table L_A (therefore $t.A$ must be invalid, since all valid values are in L_A); this has probability: pr($Err^{t.A} \bigwedge Invalid^{t.A}$) = pr($Err^{t.A}$) $\times$ pr($Invalid^{t.A}|Err^{t.A}$) = $p \times (1-c_2)$.

False negatives occur when the error is masked (i.e., when $t.A$ is incorrect but happens to be valid, and therefore is in L_A); this has probability pr($Err^{t.A} \bigwedge Valid^{t.A}$) = pr($Err^{t.A}$) $\times$ pr($Valid^{t.A}|Err^{t.A}$) = $p \times c_2$.

Finally, in this case, there can be no false positives: every A value not in L_A is an incorrect A value.

[4] A more thorough, but complex, analysis would allow errors in the table values themselves or extra/out of date values.

In the second case (i.e., assuming a imperfect lookup table), false positives show up when $t.A$ is correct, yet the value is missing from L_A; this has probability: pr($Cor^{t.A} \bigwedge L_A^{\neg t.A}$) = $(1-p) \times s$.

For true positives, another source is possible, i.e., when an incorrect $t.A$ value is valid (due to error masking), but is accidentally missing from L_A; the total probability for true positives is therefore the one obtained in the first case plus following probability: pr($Err^{t.A} \bigwedge Valid^{t.A} \bigwedge L_A^{\neg t.A}$) = pr($Err^{t.A} \bigwedge Valid^{t.A}$) $\times$ pr($L_A^{\neg t.A}$) = $(p \times c_2) \times s$.

For false negatives, we need to multiply the probability obtained in the first case by $(1-s)$, since they require the masking values also be in L_A .

Step 4: formulating assessability scores. Given the probabilities we obtained in Step 3, the expected assessability scores for $T_{B \neq A}$ can be calculated in the same way as for the case of $T_{B \neq A}$. See Table 2 (Section 4.2) for the resulting effectiveness formulas for T_{L_A}.

4.2 What-If Analysis

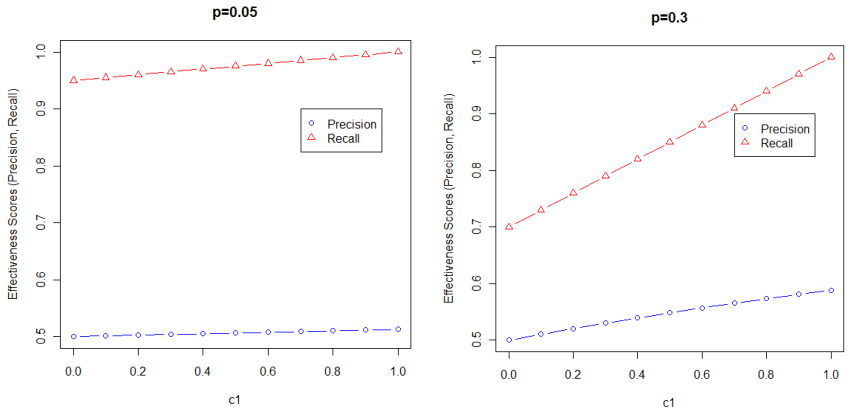
The results of the formal approach are formulas representing the expected assessability scores for DQ techniques. These formulas are useful for several reasons. First, they identify conditions (e.g., parameters p and s in Table 2) that affect the effectiveness of a DQ technique. Second, as we show below, they allow us to perform trade-off analysis concerning different scenarios that involve one or more DQ techniques. (Each scenario produces a plot of effectiveness scores by fixing most parameters and allowing the others to vary.)

The formulas that represent expected precision, recall and F-measure (when $\beta = 1$) for the DQ techniques $T_{B \neq A}$ and T_{L_A}, together with a summary of the parameters used in these formulas, are shown in Table 2. In what follows, we first show how these two techniques are evaluated individually (in Scenarios 1 - 4) and then show how they are compared with each other (in Scenarios 5 - 10).

Scenarios 1 - 4: Evaluating Individual DQ Techniques. Scenarios 1 and 2 consider the impact of "error masking" (varying c_1) on the effectiveness of $T_{B \neq A}$, while Scenarios 3

Table 2. Expected assessability scores for $T_{B \neq A}$ and T_{L_A}

Technique: $T_{B \neq A}$	***Technique***: T_{L_A}
Assessability Scores:	***Assessability Scores***:
$precision(T_{B \neq A}, A) = \frac{1+(c_1-1)p}{2+(c_1-2)p}$	$precision(T_{L_A}, A) = \frac{1+(s-1)c_2}{s/p+(s-1)(c_2-1)}$
$recall(T_{B \neq A}, A) = 1 + (c_1 - 1)p$	$recall(T_{L_A}, A) = 1 + (s-1)c_2$
$F_1(T_{B \neq A}, A) = \frac{2+2(c_1-1)p}{3+(c_1-2)p}$	$F_1(T_{L_A}, A) = \frac{2+2(s-1)c_2}{1+s/p+(s-1)(c_2-1)}$
Parameters:	
p: the probability that an A value is erroneous	
c_1: the probability that both A and B values in a tuple are erroneous, but contain different errors	
c_2: the probability that an erroneous A value is valid in the domain of A	
s: the probability that a valid A (with or without error) is not contained in the lookup table L_A	



(a) Scenario 1: a relative clean database (b) Scenario 2: a dirty database

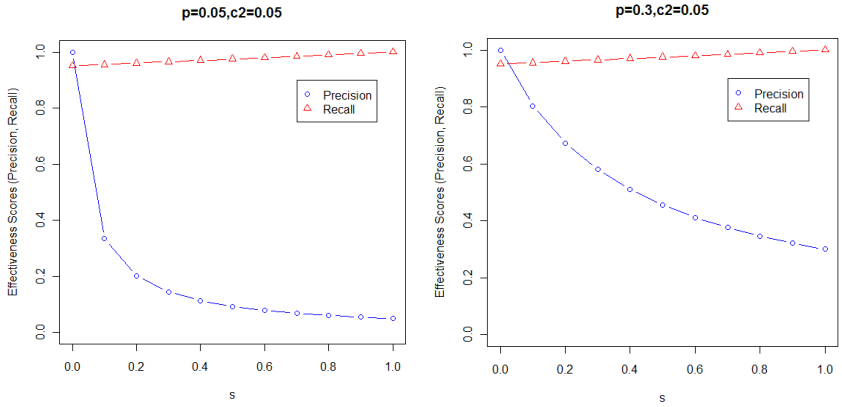
Fig. 1. Evaluation of $T_{B \neq A}$

and 4 consider the impact of L_A's "coverage" (varying s) on the effectiveness of T_{L_A}. For each technique, the evaluation is carried out with respect to a relatively clean database ($p = 0.05$, in Scenarios 1 and 3) and a dirty database, ($p = 0.3$, in Scenarios 2 and 4).

The results for Scenarios 1 and 2, as given in Figure 1(a) and 1(b), show that the precision and recall of $T_{B \neq A}$ decrease when the chance of "error masking" increases (i.e., as c_1 decreases). This corresponds to our intuition. However a comparison of these two figures also reveals that, in a dirty database (i.e., with a larger p), the effectiveness of $T_{B \neq A}$ decreases *more precipitously* as the chance of "error masking" increases. For example, as c_1 decreases from 1 to 0, the recall of $T_{B \neq A}$ decreases by only 0.05 in the clean database, but by 0.3 in the dirty database.

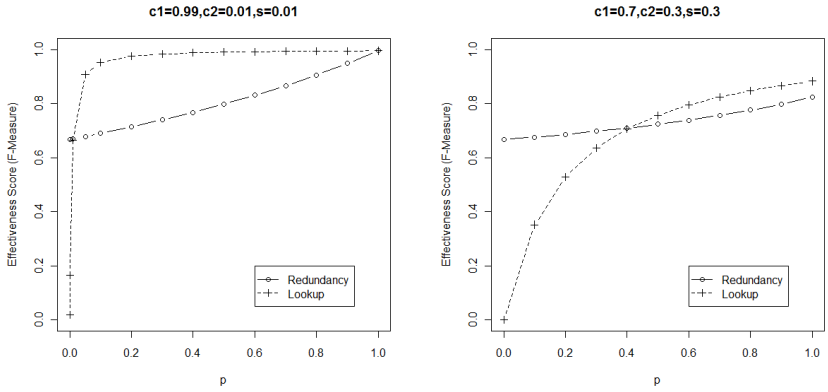
The results for Scenarios 3 and 4 are shown in Figure 2(a) and 2(b) respectively. In both cases, as the "coverage" of the lookup table decreases (i.e., as s increases), we notice an intuitively expected decrease in precision; however, the dramatic nature of its drop is not so easily predicted by intuition, and is therefore a benefit of this analysis. We also note that recall is much less affected by the "coverage". Moreover, by comparing these two figures, we observe that the probability of errors in A has much greater impact on precision than on recall. More specifically, the recall of T_{L_A} remains the same when comparing the clean and dirty databases; however, in the dirty database, the precision decreases considerably slower as the "coverage" decreases. For example, when s increases from 0 to 1, the precision of T_{L_A} decreases by 0.95 in the clean database, and by only 0.7 in the dirty database.

Scenarios 5 and 6: Comparing DQ Techniques - The Impact of Errors. In this subsection, we compare DQ techniques $T_{B \neq A}$ and T_{L_A} in two scenarios, by investigating the impact of the probability of errors in A (varying p) on the effectiveness of these two techniques in an optimistic and a pessimistic setting. In the optimistic case, the chance of "error masking" is very small and the "coverage" of the lookup table is nearly perfect. More specifically, we assume that (i) in 99% of the cases, erroneous A and B values in a tuple contain different errors (i.e., $c_1 = 0.99$), (ii) only 1% of erroneous A



(a) Scenario 3: a relative clean database (b) Scenario 4: a dirty database

Fig. 2. Evaluation of T_{L_A}



(a) Scenario 5: an optimistic view (b) Scenario 6: a pessimistic view

Fig. 3. Comparison of $T_{B \neq A}$ and T_{L_A} - Impact of Errors

values happen to be other valid values (i.e., $c_2 = 0.01$), and (iii) only 1% of the valid A values are not contained in the lookup table L_A (i.e., $s = 0.01$). In the pessimistic case, we significantly increase the chance of "error masking" and decrease the "coverage" of the lookup table. More specifically, we set $c_1 = 0.70$, $c_2 = 0.30$ and $s = 0.30$. Figure 3(a) and 3(b) compare the F-Measures of $T_{B \neq A}$ and T_{L_A} in these scenarios.

We observe that in both settings, the F-measure of $T_{B \neq A}$ increases as the number of erroneous A values increases (i.e., p increases). A similar pattern can be observed for T_{L_A} in the pessimistic setting; in the optimistic setting, the F-measure of T_{L_A} increases dramatically when $p < 0.05$, and remains almost constant when $p \geq 0.05$. These two figures suggest under what circumstances one DQ technique is preferable to the other one. More specifically, in an optimistic world, $T_{B \neq A}$ outperforms T_{L_A} only when the probability of erroneous A values is quite small (i.e., when $p < 0.01$), while in a

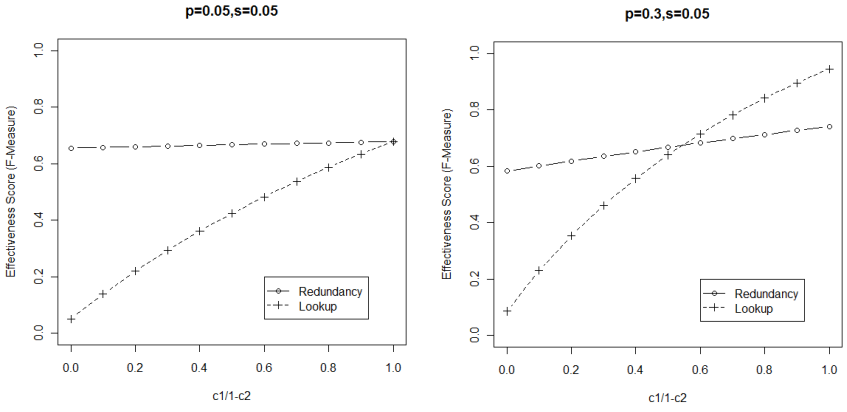
pessimistic world, $T_{B \neq A}$ is a more effective choice than T_{L_A} as long as the error rate in A is less than 40% (i.e., when $p < 0.4$).

A briefer summary might be that in a typical situation, where the chance of "error masking" is reasonably small (say, less than 5%) and the "coverage" of lookup table is nearly perfect (say, more than 95%), a lookup-table based DQ technique is generally more effective in detecting errors than a redundancy-check based DQ technique, as long as the database is expected to have more than 5% erroneous values.

Scenarios 7 - 10: Comparing DQ Techniques - The Impact of "Error Masking". In this subsection, we compare the DQ techniques $T_{B \neq A}$ and T_{L_A} in another four scenarios, by investigating the impact of the "error masking" (varying c_1 and c_2) on the effectiveness of these techniques. The comparison is carried out with respect to a relatively clean database, i.e., $p = 0.05$ (Scenarios 7 and 9) and a dirty database, i.e., $p = 0.30$ (Scenarios 8 and 10), and as well as with respect to a nearly perfect lookup table, i.e., s = 0.01 (Scenarios 7 and 8), and an imperfect lookup table, i.e., $s = 0.3$ (Scenarios 9 and 10).

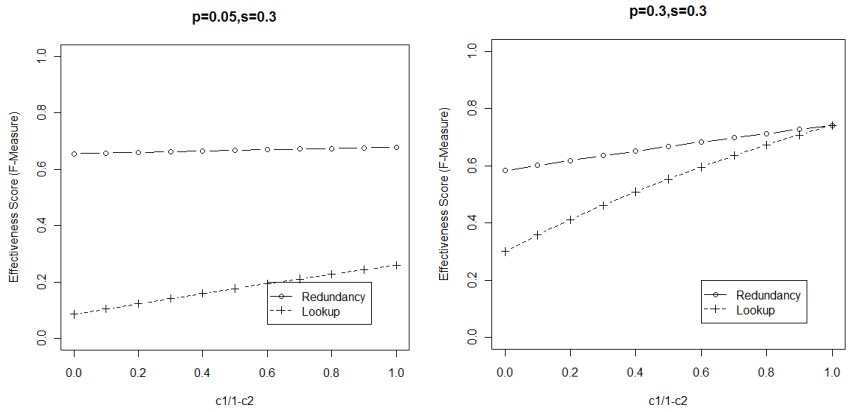
Figure 4(a) and 4(b) compare the F-Measures of $T_{B \neq A}$ and T_{L_A} in Scenarios 7 and 8, while Figure 5(a) and 5(b) compare them in Scenarios 9 and 10. From these figures, a dominant pattern can be observed: the chance of "error masking" has more impact on T_{L_A} than on $T_{B \neq A}$, and this influence is independent of the probabilities of errors in A and the "coverage" of L_A. In other words, the F-measure of T_{L_A} increases more precipitously than that of $T_{B \neq A}$ does (in all four cases) as the chance of "error masking" decreases (i.e., as c_1 and $1 - c_2$ increases).

In addition to this general pattern, the following conclusion can be reached according to these figures. For relative clean databases with less than 5% of erroneous values, a redundancy-check based DQ technique is always more effective than a lookup-table based technique. When there are more than 5% of erroneous values, the redundancy-check based technique still outperforms the lookup-table based one, unless a relatively low chance of "error masking" is guaranteed and the "coverage" of the lookup table is



(a) Scenario 7: clean db, good lookup (b) Scenario 8: dirty db, good lookup

Fig. 4. Comparison of $T_{B \neq A}$ and T_{L_A} - Impact of "Error Masking" (I)



(a) Scenario 9: clean db, bad lookup (b) Scenario 10: dirty db, bad lookup

Fig. 5. Comparison of $T_{B \neq A}$ and T_{L_A} - Impact of "Error Masking" (II)

nearly perfect. This conclusion, together with the one we made in Scenarios 5 and 6, gives us a complete comparison of $T_{B \neq A}$ and T_{L_A}.

The general point is that the mathematical assessment of the effectiveness of DQ techniques based on probabilistic parameters allows us to make judgments about when to use one technique vs. another, or whether to use one at all – we need to remember that there is an overhead for putting into place a DQ technique.

5 Related Work

Software engineering researchers and practitioners have been developing and using numerous metrics for assessing and improving quality of software and its development processes [8]. In comparison, measures for DQ and DQ techniques have received less attention. Nevertheless, significant amount of effort has been dedicated to the classification and definition of DQ dimensions. Each such dimension aims at capturing and representing a specific aspect of quality in data, and can be associated with one or more measures according to different factors involved in the measurement process [1]. Measures for accuracy, completeness, timeliness dimensions have been proposed in [9,10,11].

Although to the best of our knowledge no general measurement framework exists for DQ techniques, performance measures have been proposed and used for certain types of techniques (such as Record linkage). Performance measures in this case are often defined as the functions of the number of true positives, false positives, etc [12,13]. For example, in addition to precision, recall and F-measures, the performance of a record linkage algorithm can also be measured using *Accuracy = (TP + TN) / (TP + FP + TN + FN)*, among others [13]. In these proposals, performance scores are obtained for particular applications of a record linkage algorithm on actual data sets, and are mainly used as a mechanism to tune the parameters (e.g., matching threshold) of the algorithm. The present paper focuses on the estimation of expected effectiveness scores and the comparison of DQ techniques under different scenarios. The formal techniques and what-if

analysis presented in this paper are therefore complementary to the existing performance measures used for record linkage algorithms.

As we have discussed, the schema of a database plays a significant role in ensuring quality of data in the database. Researchers in conceptual modeling have worked on the understanding and characterization of quality aspects of schemas, such as (schema) completeness, minimality, pertinence [14,1]. Moreover, quality measures have been proposed for ER schemas; for example, the integrity measure in [15] is defined using the number of incorrect integrity constraints and the number of correct ones that are not enforced. Quality measures for logical schemas have also been developed in [16,17,18,19]. The question that remains is — if quality of schema influences that of data, how is this influence reflected in their quality measures. The present paper can be seem as one step toward answering this question. Since DQ techniques rely on changes to the structure and elements of schemas (and the specifiable constraints according the changes), their effectiveness measures contribute to the measurement of schemas′ *controllability* on DQ problems, another quality measure for schemas yet to be explored. Our effectiveness measures therefore help us to understand the relationship between the ability to control DQ problems at the schema level and the actual manifestation of these problems at instance level.

6 Conclusion

In this paper, we have proposed a quantitative approach for measuring the effectiveness of DQ techniques. Inspired by Information Retrieval, we started by proposing to calculate *numeric effectiveness scores* for a DQ technique by comparing its performance on a database instance with that of humans, who are assumed to have perfect knowledge of the world represented by that instance. As in Information Retrieval, this has the weakness of depending on the particular database instance used, and may require significant human effort in evaluating the actual data.

We therefore generalized the idea by introducing probabilistic assumptions concerning the occurrence of errors in data values and confounding factors that may render the DQ technique less effective. These assumptions are expressed in terms of probability distributions for various events, each characterized by certain parameters. We then showed with several examples how one can obtain *mathematical formulas* for the effectiveness of a DQ technique, which involve the parameters of the above-mentioned distributions. This is a significant advance, because it provides a way for the effectiveness of a DQ technique to be evaluated over a range of possible values for the parameters. This allows us for the first time to compare in a mathematically precise way different DQ techniques, and talk about the circumstances when one becomes better than another. Moreover, it lays the foundations for future research on optimal allocation of resources for DQ enforcement.

Ongoing and future work is needed to fulfill the promise of this approach. This includes to identify classes of DQ techniques (integrity constraints and workflows) for which the formal approach for deriving effectiveness formulas, as illustrated in this paper, can be mechanized or at least reduced to a systematic methodology. This will likely include a comprehensive classification of DQ techniques, and will result in a library of DQ techniques augmented by their effectiveness formulas. The next stage is to also make

the process of generating interesting scenarios more systematic. Finally, while this paper concentrated only on error detection, there are many other aspects, such as error correction and monitoring, that can be brought into this more precise, mathematical approach.

References

1. Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications), 1st edn. Springer, Heidelberg (2006)
2. Jiang, L., Topaloglou, T., Borgida, A., Mylopoulos, J.: Goal-oriented conceptual database design. In: Proceedings of the 15th IEEE International Requirements Engineering Conference (RE 2007) (2007)
3. Jiang, L., Borgida, A., Topaloglou, T., Mylopoulos, J.: Data quality by design: A goal-oriented approach. In: Proceedings of the 12th International Conference on Info. Quality (ICIQ 2007) (2007)
4. Bohannon, P., Fan, W., Geerts, F., Jia, X., Kementsietsidis, A.: Conditional functional dependencies for data cleaning. In: IEEE 23rd International Conference on Data Engineering, 2007. ICDE 2007, pp. 746–755 (2007)
5. Fan, W., Geerts, F., Jia, X., Kementsietsidis, A.: Conditional functional dependencies for capturing data inconsistencies. ACM Trans. Database Syst. 33(2), 1–48 (2008)
6. van Rijsbergen, C.: Information Retrieval, 2nd edn. Butterworth, London (1979)
7. Barbará, D., Goel, R., Jajodia, S.: Using checksums to detect data corruption. In: Advances in Database Technology — EDBT 2000, pp. 136–149 (2000)
8. Fenton, N.E., Pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach. PWS Publishing Co., Boston (1998)
9. Ballou, D., Wang, R., Pazer, H., Tayi, G.K.: Modeling information manufacturing systems to determine information product quality. Manage. Sci. 44(4), 462–484 (1998)
10. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Communications of the ACM 45(4), 211–218 (2002)
11. Ballou, D.P., Pazer, H.L.: Modeling completeness versus consistency tradeoffs in information decision contexts. IEEE Trans. on Knowl. and Data Engineering 15(1), 240–243 (2003)
12. Gu, L., Baxter, R., Vickers, D., Rainsford, C.: Record linkage: Current practice and future directions. Technical report, CSIRO Mathematical and Information Sciences (2003)
13. Christen, P., Goiser, K.: Quality and complexity measures for data linkage and deduplication. In: Guillet, F., Hamilton, H.J. (eds.) Quality Measures in Data Mining. Studies in Computational Intelligence, vol. 43, pp. 127–151. Springer, Heidelberg (2007)
14. Batini, C., Ceri, S., Navathe, S.B.: Conceptual Database Design: An Entity-Relationship Approach. Benjamin/Cummings (1992)
15. Moody, D.L.: Metrics for evaluating the quality of entity relationship models. In: Ling, T.-W., Ram, S., Li Lee, M. (eds.) ER 1998. LNCS, vol. 1507, pp. 211–225. Springer, Heidelberg (1998)
16. Piattini, M., Calero, C., Genero, M.: Table oriented metrics for relational databases. Software Quality Journal 9(2), 79–97 (2001)
17. Calero, C., Piattini, M.: Metrics for databases: a way to assure the quality. In: Piattini, M.G., Calero, C., Genero, M. (eds.) Information and database quality, pp. 57–84. Kluwer Academic Publishers, Norwell (2002)
18. Baroni, A.L., Calero, C., Abreu, F.B., Piattini, M.: Object-relational database metrics formalization. In: Sixth International Conference on Quality Software, pp. 30–37. IEEE Computer Society, Los Alamitos (2006)
19. Serrano, M.A., Calero, C., Piattini, M.: Metrics for data warehouse quality. In: Khosrow-Pour, M. (ed.) Encyclopedia of Info. Sci. and Techno. (IV), pp. 1938–1944. Idea Group (2005)