ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

Creating Adaptive Feedback Designed for Improving Data Entry Accuracy

K. Nagendra Rao[1], K.Venkata Ramana[2]

Associate Professor, Dept. of CSE, CR Engineering College, Guntur, Andhra Pradesh, India

Research Scholar, Dept. of CSE, Dr.RR &Dr.SR Technological University, Chennai, Tamil Nadu, India

ABSTRACT: Data quality is a critical problem in modern databases. Data-entry forms present the first and arguably best opportunity for detecting and mitigating errors, but there has been little research into automatic methods for improving data quality at entry time. In this paper, we propose USHER, an end-to-end system for form design, entry, and data quality assurance. Using previous form submissions, USHER learns a probabilistic model over the questions of the form. USHER then applies this model at every step of the data-entry process to improve data quality. Before entry, it induces a form layout that captures the most important data values of a form instance as quickly as possible and reduces the complexity of error-prone questions. During entry, it dynamically adapts the form to the values being entered by providing real-time interface feedback, reasking questions with dubious responses, and simplifying questions by reformulating them. After entry, it revisits question responses that it deems likely to have been entered incorrectly by reasking the question or a reformulation thereof. We evaluate these components of USHER using two real-world data sets. Our results demonstrate that USHER can improve data quality considerably at a reduced cost when compared to current practice.

KEYWORDS: Data quality, data entry, form design, adaptive form.

1. INTRODUCTION

Organizations and individuals routinely make important decisions based on inaccurate data stored in supposedly authoritative databases. Data errors in some domains, such as medicine, may have particularly severe consequences. These errors can arise at a variety of points in the life cycle of data, from data entry, through storage, integration, and cleaning, all the way to analysis and decision making [1].While each step presents an opportunity to address data quality, entry time offers the earliest opportunity to catch and correct errors. The database community has focused on data cleaning once data have been collected into a database, and has paid relatively little attention to data quality at collection time [1], [2]. Current best practices for quality during data entry come from the field of survey methodology, which offers principles that include manual question orderings and input constraints, and double entry of paper forms [3].Although this has long been the de facto quality assurance standard in data collection and transformation, we believe this area merits reconsideration. For both paper forms and direct electronic entry, we posit that a data-driven and more computationally sophisticated approach can significantly. Outperform these decades-old static methods in both accuracy and efficiency of data entry.

The problem of data quality is magnified in low-resource data collection settings. Recently, the World Health Organization likened the lack of quality health information in developing regions to a "gathering storm," saying, "[to] make people count, we first need to be able to count people"[4]. Indeed, many health organizations, particularly those operating with limited resources in developing regions, struggle with collecting high-quality data. Why is data collection so challenging? First, many organizations lack expertise in paper and electronic form design: designers approach question and answer choice selection with a defensive, catchall mind-set, adding answer choices and questions that may not be necessary; furthermore, they engage in ad hoc mapping of required data fields to data entry widgets by intuition [5], [6], often ignoring or specifying ill-fitting constraints. Second, double entry is too costly. In some cases this means it is simply not performed, resulting in poor data quality. In other cases, particularly when double entry is mandated by third

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

parties, it results in delays and other unintended negative consequences. We observed this scenario in an HIV/AIDS program in Tanzania, where time-consuming double entry was imposed upon a busy local clinic. The effort required to do the double entry meant that the transcription was postponed for months and handled in batch. Although the data eventually percolated up to national and international agencies, in the interim the local clinic was operating as usual via paper forms, unable to benefit from an electronic view of the data latent in their organization. Finally, many organizations in developing regions are beginning to use mobile devices like smartphones for data collection; for instance, community health workers are doing direct digital data entry in remote locations. Electronic data-entry devices offer different affordances than those of paper, displacing the role of traditional form design and double entry [5]. We often found that there were no data quality checks at all in aı¨vely implemented mobile interfaces, compounding the fact that mobile data-entry quality is 10 times worse than dictation to a human operator [7].

In addition, we may extend USHER's appropriate entry friction approach to provide a framework for reasoning about feedback mechanisms for the data-entry user interface.During data entry, using the likelihood of unanswered fields given entered answers, and following the intuition that multivariate outliers are values warranting reexamination by the data-entry worker, USHER can guide the user with much more specific and context-aware feedback. In Section 9, we offer initial thoughts on design patterns for USHER-inspired dynamic data-entry interfaces.The contributions of this paper are fourfold:

1. We describe the design of USHER's core: probabilistic models for arbitrary data-entry forms.
2. We describe USHER's application of these models to provide guidance along each step of the data-entry lifecycle: reordering questions for greedy information gain, reformulating answers for appropriate entry friction, and reasking questions according to contextualized error likelihood.
3. We present experiments showing that USHER has the potential to improve data quality at reduced cost. We study two representative data sets: direct electronic entry of survey results about political opinion and transcription of paper-based patient intake forms from an HIV/AIDS clinic in Tanzania.
4. Extending our ideas on form dynamics, we propose new user-interface principles for providing contextualized, intuitive feedback based on the likelihood of data as they are entered. This provides a foundation for incorporating data cleaning mechanisms directly in the entry process.

II. RELATED WORK

2.1. Data Cleaning

In the database literature, data quality has typically been addressed under the rubric of data cleaning [1], [2]. Our work connects most directly to data cleaning via multivariate outlier detection; it is based in part on interface ideas first proposed by Hellerstein [8]. By the time such retrospective data cleaning is done, the physical source of the data is typically unavailable—thus, errors often become too difficult or time-consuming to be rectified. USHER addresses this issue by applying statistical data quality insights at the time of data entry. Thus, it can catch errors when they are made and when ground-truth values may still be available for verification.

2.2. User Interfaces

Past research on improving data entry is mostly focused on adapting the data-entry interface for user efficiency improvements. Several such projects have used learning techniques to automatically fill or predict a top-k set of likely values [9], [10], [11], [12], [13], [14], [15]. For example, Ali and Meek [9] predicted values for combo-boxes in web forms and measured improvements in the speed of entry, Ecopod [15] generated type-ahead suggestions that were improved by geographic information, and Hermens and Schlimmer [10] automatically filled leave of absence forms using decision trees and measured predictive accuracy and time savings. In these approaches, learning techniques are used to predict form values based on past data, and each measures the time savings of particular data-entry mechanisms and/or the proportion of values their model was able to correctly predict. USHER's focus is on improving data quality, and its probabilistic formalism is based on learning relationships within the underlying data that guide the user toward correct entries. In addition to predicting question values, we develop and exploit probabilistic models of user error, and target a broader set of interface adaptations for improving data quality, including question reordering, reformulation, and reask-

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

ing, and widget customizations that provide feedback to the user based on the likelihood of their entries. Some of the enhancements we make for data quality could also be applied to improve the speed of entry.

2.3. Clinical Trials

Data quality assurance is a prominent topic in the science of clinical trials, where the practice of double entry has been questioned and dissected, but nonetheless remains the gold standard [16], [17]. In particular, Kleinman takes a probabilistic approach toward choosing which forms to reenter based on the individual performance of data-entry staff [18]. This cross-form validation has the same goal as our approach of reducing the need for complete double entry, but does so at a much coarser level of granularity. It requires historical performance records for each data-entry worker, and does not offer dynamic reconfirmation of individual questions. In contrast, USHER's cross-question validation adapts to the actual data being entered in light of previous form submissions, and allows for a principled assessment of the trade-off between cost (of reconfirming more questions) versus quality (as predicted by the probabilistic model).

2.4. Survey Design

The survey design literature includes extensive work on form design techniques that can improve data quality [3], [19]. This literature advocates the use of manually specified constraints on response values. These constraints may be univariate (e.g., a maximum value for an age question) or multivariate (e.g., disallowing gender to be male and pregnant to be yes). Some constraints may also be "soft" and only serve as warnings regarding unlikely combinations (e.g., age being 60 and pregnant being yes).

The manual specification of such constraints requires a domain expert, which can be prohibitive in many scenarios. By relying on prior data, USHER learns many of these same constraints without requiring their explicit specification.When these constraints are violated during entry,USHER can then flag the relevant questions, or target them for reasking.
However, USHER does not preclude the manual specification of constraints. This is critical, because previous research into the psychological phenomena of survey filling has yielded common constraints not inherently learnable from prior data [3]. This work provides heuristics such as "groups of topically related questions should often be placed together" and "questions about race should appear at the end of a survey." USHER complements these humanspecified constraints, accommodating them while leveraging any remaining flexibility to optimize question ordering in a data-driven manner.

III. SYSTEM

USHER builds a probabilistic model for an arbitrary dataentry form in two steps: first, by learning the relationships between form questions via structure learning, resulting in a Bayesian network; and second, by estimating the parameters of that Bayesian network, which then allows us to generate predictions and error probabilities for the form. After the model is built, USHER uses it to automatically order a form's questions for greedy information gain. Section 5 describes both static and dynamic algorithms that employ criteria based on the magnitude of statistical information gain that is expected in answering a question, given the answers that have been provided so far. This is a key idea in our approach. By front-loading predictive potential, we increase the models' capacity in several ways. First, from an information-theoretic perspective, we improve our ability to do multivariate prediction and outlier detection for subsequent questions. As we discuss in more detail in Sections 7 and 9, this predictive ability can be applied by reformulating error-prone form questions, parameterizing data-entry widgets (type-ahead suggestions and default values), assessing answers (outlier flags), and performing in-flight reasking (also known as cross validation in survey design parlance). Second, from a psychological perspective, frontloading information gain also addresses the human issues of user fatigue and limited attention span, which can result in increasing error rates over time and unanswered questions at the end of the form.

Our approach is driven by the same intuition underlying the practice of curbstoning, which was related to us in discussion with survey design experts [6]. Curbstoning is a way in which an unscrupulous door-to-door surveyor shirks work: he or she asks an interviewee only a few important questions, and then uses those responses to complete the remainder of a form while sitting on the curb outside the home. The constructive insight here is that a well-chosen subset of ques-

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

tions can often enable an experienced agent to intuitively predict the remaining answers. USHER's question ordering algorithms formalize this intuition via the principle of greedy information gain, and use them (scrupulously) to improve data entry.

USHER's learning algorithm relies on training data. In practice, a data-entry backlog can serve as this training set. In the absence of sufficient training data, USHER can bootstrap itself on a "uniform prior," generating a form based on the assumption that all inputs are equally likely; this is no worse than standard practice. Subsequently, a training set can gradually be constructed by iteratively capturing data from designers and potential users in "learning runs." It is a common approach to first fit to the available data, and then evolve a model as new data become available. This process of semiautomated form design can help institutionalize new forms before they are deployed in production.

USHER adapts to a form and data set by crafting a custom model. Of course, as in many learning systems, the model learned may not translate across contexts. We do not claim that each learned model would or should fully generalize to different environments. Instead, each contextspecific model is used to ensure data quality for a particular situation, where we expect relatively consistent patterns in input data characteristics. In the remainder of this section, we illustrate USHER's functionality with examples. Further details, particularly regarding the probabilistic model, follow in the ensuing sections.

3.1. Examples

We present two running examples. First, the patient comes from paper patient-registration forms transcribed by data-entry workers at an HIV/AIDS program in Tanzania.1 Second, the survey data set comes from a phone survey of

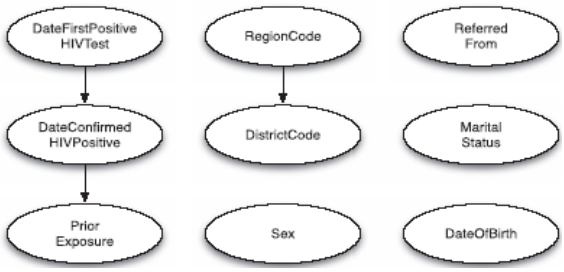


Fig.1. *Bayesian network for the patient data set,showing Automatically inferred probabilistic relationships Between form questions.*

political opinion in the San Francisco Bay Area, entered by survey professionals directly into an electronic form.
In each example, a form designer begins by creating a simple specification of form questions and their prompts,response data types, and constraints. The training data set is made up of prior form responses. Using the learning algorithms we present in Section 4, USHER builds a Bayesian network of probabilistic relationships from the data, as shown in Figs. 1 and 2. In this graph, an edge captures a close stochastic dependency between two random variables (i.e.,form questions). Two questions with no path between them in the graph are probabilistically independent. Fig. 2 illustrates a denser graph, demonstrating that political survey responses tend to be highly correlated. Note that a standard joint distribution would show correlations among all pairs of questions; the sparsity of these examples reflects conditional independence patterns learned from the data.Encoding independence in a Bayesian network is a standard method in machine learning that clarifies the underlying structure, mitigates data overfitting, and improves the efficiency of probabilistic inference.

The learned structure is subject to manual control: a designer can override any learned correlations that are believed to be spurious or that make the form more difficult to administer.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

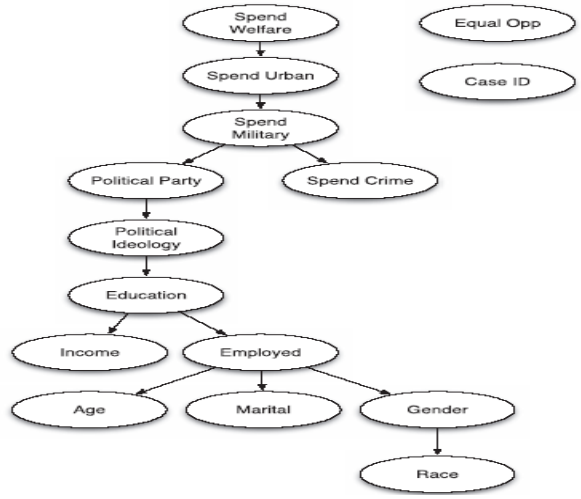


Fig.2. *Bayesian network for the survey data set.*

static ordering shown in Fig. 3. We can see in Fig. 3 that the structure learner predicted RegionCode to be correlated with DistrictCode. Our data set is collected mostly from clinics in a single region of Tanzania, so RegionCode provides little information. It is not surprising then, thatUSHER's suggested ordering has DistrictCode early and RegionCode last—once we observe DistrictCode, RegionCode has very little additional expected conditional information gain. When it is time to input the RegionCode, if the user selects an incorrect value, the model can be more certain that it is unlikely. If the user stops early and does not fill in RegionCode, the model can infer the likely value with higher confidence. In general, static question orderings are appropriate as an offline process for paper forms where there is latitude for (re)ordering questions, within designer-specified constraints.

During data entry, USHER uses the probabilistic machinery to drive dynamic updates to the form structure. One type of update is the dynamic selection of the best next question to ask among questions yet to be answered. This can be appropriate in several situations, including surveys that do not expect users to finish all questions, or direct-entry interfaces (e.g., mobile phones) where one question is asked at a time. We note that it is still important to respect the form designer's a priori specified question-grouping and –ordering constraints when a form is dynamically updated. USHER is also used during data entry to provide dynamic feedback, by calculating the conditional distribution for the question in focus and using it to influence the way the question is presented. We tackle this via two techniques: question reformulation and widget decoration. For the former, we could, for example, choose to reformulate the question about RegionCode into a binary yes/no

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

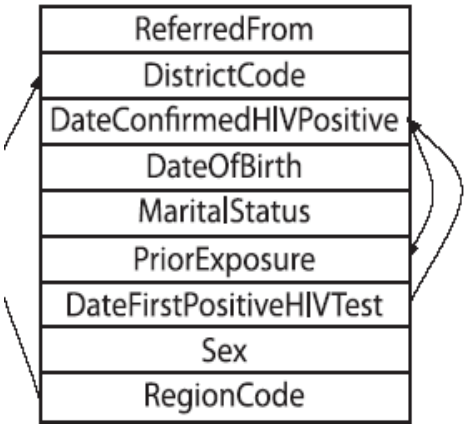
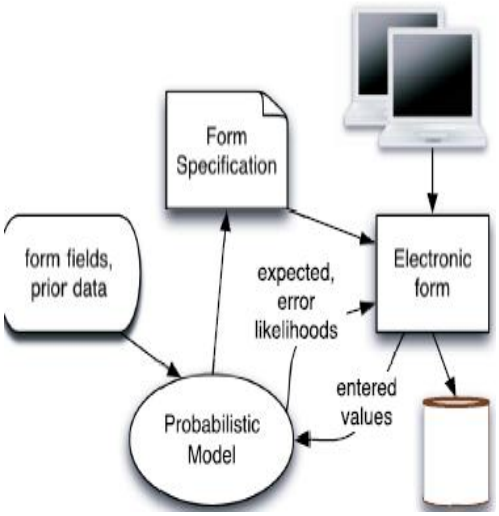


Fig. 3. *Example question layout generated by our ordering algorithm. The arrows reflect the probabilistic dependencies from Fig. 1.*

question based on the answer to DistrictCode, since DistrictCode is such a strong predictor of RegionCode. As we discuss in Section 7, the reduced selection space for responses in turn reduces the chances of a data-entry worker selecting an incorrect response. For the latter, possibilities include using a "split" drop-down menu for RegionCode that features the most likely answers "above the line," and after entry, coloring the chosen answer red if it is a conditional outlier. We discuss in Section 9 the design space and potential impact of data-entry feedback that is more specific and context-aware.

As a form is being filled, USHER calculates contextualized error probabilities for each question. These values are used for reasking questions in two ways: during primary form entry and for reconfirming answers after an initial pass. For each form question, USHER predicts how likely the response provided is erroneous, by examining whether it is likely to be a multivariate outlier, i.e., that it is unlikely with respect to the responses for other fields. In other words, an error probability is conditioned on all answered values provided by the data-entry worker so far



ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

3.2.Implementation

We have implemented USHER as a web application (Fig. 4). The UI loads a simple form specification file containing form question details and the location of the training data set. Form question details include question name, prompt, data type,widget type, and constraints. The server instantiates a model for each form. The system passes information about question responses to the model as they are filled in; in exchange, the model returns redictions and error probabilities.Models are created from the form specification, the training data set, and a graph of learned structural relationships.We perform structure learning offline with BANJO [20], an open source Java package for structure learning of Bayesian networks. Our graphical model is implemented in two variants: the first model used for ordering is based on a modified version of JavaBayes [21], an open source Java software for ayesian inference. Because JavaBayes only supports discrete probability variables, we implemented the error prediction version of our model using Infer.NET [22], a Microsoft .NET Framework toolkit for Bayesian inference.

VI. LEARNING A MODEL FOR DATA ENTRY

The core of the USHER system is its probabilistic model of the data, represented as a Bayesian network over form questions. This network captures relationships between a form's question elements in a stochastic manner. In particular, given input values for some subset of the questions of a particular form instance, the model can infer probability distributions over values of that instance's remaining unanswered questions. In this section, we show how standard machine learning techniques can be used to induce this model from previous form entries.

We will use F={F1, . . . ; Fn} to denote a set of random variables representing the values of n questions comprising a data-entry form. We assume that each question response takes on a finite set of discrete values; continuous values are discretized by dividing the data range into intervals and assigning each interval one value.2 To learn the probabilistic model, we assume access to prior entries for the same form.

USHER first builds a Bayesian network over the form questions, which will allow it to compute probability distributions over arbitrary subsets G _ F of form question random variables, given already entered question responses G` = g` for that instance, i.e., P(G|G`=g`).Constructing this network requires two steps: first, the induction of the graph structure of the network, which encodes the conditional independencies between the question random variables F; and second, the estimation of the resulting network's parameters.

The naive approach to structure selection would be to assume complete dependence of each question on every other question. However, this would blow up the number of free parameters in our model, leading to both poor generalization performance of our predictions and prohibitively slow model queries. Instead, we learn the structure using the prior form submissions in the database. USHER searches through the space of possible structures using simulated annealing, and chooses the best structure according to the Bayesian Dirichlet Equivalence criterion [23]. This criterion optimizes for a trade-off between model expressiveness (using a richer dependency structure) and model parsimony (using a smaller number of parameters), thus identifying only the prominent, recurring probabilistic dependencies.Fig. 1 and 2 show automatically learned structures
For two data domains.

V. DISCUSSION AND FUTURE WORK

In this paper, we have shown that a probabilistic approach can be used to design intelligent data-entry forms that promote high data quality. USHER leverages data-driven insights to automate multiple steps in the data-entry pipeline. Before entry, we find an ordering of form fields that promotes rapid information capture, driven by a greedy information gain principle, and can statically reformulate questions to promote more accurate responses. During entry, we dynamically adapt the form based on entered values, facilitating reasking, reformulation, and real-time interface feedback in the sprit of providing appropriate entry friction. After entry, we automatically identify possibly erroneous inputs, guided by contextualized error likelihoods, and reask those questions, possibly reformulated, to verify their correctness.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 11, September 2015

National Conference on Emerging Trends in Computing, Communication & Control Engineering [NCET 2K15]

On 4th September, 2015

Organized by

Department of CSE, ECE & EEE, Magna College of Engineering, Chennai-600055, India.

Our simulated empirical evaluations demonstrate the data quality benefits of each of these components: question ordering, reformulation and reasking.The USHER system we have presented is a cohesive synthesis of several disparate approaches to improving data quality for data entry. The three major components of the system—ordering, reasking, and reformulation—can all be applied under various guises before, during, and after data entry. This suggests a principled road map for future research in data entry. For example, one combination we have not explored here is reasking before entry. At first glance this may appear strange, but in fact that is essentially the role that cross-validation questions in paper forms serve, as preemptive reformulated reasked questions. Translating such static cross-validation questions to dynamic forms is a potential direction of future work. Another major piece of future work alluded to in Section 9 is to study how our probabilistic model can inform effective adaptations of the user interface during data entry. We intend to answer this problem in greater depth through user studies and field deployments of our system. We can also extend this work by enriching the underlying probabilistic formalism. Our current probabilistic approach assumes that every question is discrete and takes on a series of unrelated values. Relaxing these assumptions would make for a potentially more accurate predictive model for many domains. Additionally, we would want to consider models that reflect temporal changes in the underlying data. Our present error model makes strong assumptions both about how errors are distributed and what errors look like. On that front, an interesting line of future work would be to learn a model of data-entry errors and adapt our system to catch them. Finally, we are in the process of measuring the practical impact of our system, by piloting USHER with our field partners, the United Nations Development Program's Millennium Villages Project [34] in Uganda, and a community health care program in Tanzania. These organizations' data quality concerns were the original motivation for this work and thus serve as an important litmus test for our system.

VI. ACKNOWLEDGMENTS

The authors thank Maneesh Agrawala, Yaw Anokwa, Michael Bernstein, S.R.K. Branavan, Tyson Condie, Heather Dolan, Jeff Heer, Max van Kleek, Neal Lesh, Alice Lin, Jamie Lockwood, Bob McCarthy, Tom Piazza, Christine Robson, and the anonymous reviewers for their helpful comments and suggestions. We acknowledge the Yahoo Labs Technology for Good Fellowship, the Natural Sciences and Engineering Research Council of Canada, the US National Science Foundation (NSF) Graduate Fellowship, and NSF Grant 0713661.

REFERENCES

1. T. Dasu and T. Johnson, Exploratory Data Mining and Data Cleaning. Wiley, 2003.
2. C. Batini and M. Scannapieco, Data Quality: Concepts, Methodologies and Techniques. Springer, 2006.
3. R.M. Groves, F.J. Fowler, M.P. Couper, J.M. Lepkowski, E. Singer, and R. Tourangeau, Survey Methodology. Wiley-Interscience, 2004.
4. J. Lee, "Address to WHO Staff," http://www.who.int/dg/lee/speeches/2003/21_07/en, 2003.
5. J.V.D. Broeck, M. Mackay, N. Mpontshane, A.K.K. Luabeya, M.Chhagan, and M.L. Bennish, "Maintaining Data Integrity in a Rural Clinical Trial," Clinical Trials, vol. 4, pp. 572-582, 2007.
6. R. McCarthy and T. Piazza, "Personal Interview," Univ. of California at Berkeley Survey Research Center, [7] S. Patnaik, E. Brunskill, and W. Thies, "Evaluating the Accuracy of Data Collection on Mobile Phones: A Study of Forms, SMS, and Voice," Proc. IEEE/ACM Int'l Conf. Information and Comm.Technologies and Development (ICTD), 2009.
7. J.M. Hellerstein, "Quantitative Data Cleaning for Large Databases," United Nations Economic Commission for Europe (UNECE), 2008.
8. A. Ali and C. Meek, "Predictive Models of Form Filling,"Technical Report MSR-TR-2009-1, Microsoft Research, Jan. 2009.
9. L.A. Hermens and J.C. Schlimmer, "A Machine-Learning Apprentice for the Completion of Repetitive Forms," IEEE Expert:
10. Intelligent Systems and Their Applications, vol. 9, no. 1, pp. 28-33,Feb. 1994. [11] D. Lee and C. Tsatsoulis, "Intelligent Data Entry Assistant for XML Using Ensemble Learning," Proc. ACM 10th Int'l Conf.Intelligent User Interfaces (IUI), 2005.