Data Quality Aware Query System

Naiem Khodabandehloo Yeganeh
supervised by
Shazia Sadiq
The University of Queensland
St Lucia
QLD, Australia
naiem@itee.uq.edu.au

ABSTRACT

Traditional query systems do not factor in data quality considerations in their response. However, the issue of data quality is of growing importance as individuals as well as corporations are increasingly relying on multiple, often external sources of data to make decisions. Previous studies have identified diverse interpretations of data quality indicating that fitness for use is a fundamental criterion in the evaluation of data quality. In this thesis, we propose data quality aware query systems that are designed to facilitate access to data which is fit for use. Three major challenges of such systems have been identified and are being addressed: profiling the quality of data sets, modelling user preferences on data quality, and data quality aware query planning.

1. INTRODUCTION

User satisfaction from a query response is a complex problem encompassing various dimensions including both the efficiency as well as the quality of the response. Quality in turn includes several dimensions such as completeness, currency, accuracy, relevance and many more [9]. In current information environments where individuals as well as corporations are routinely relying on multiple, external data sources for their information needs, absolute metrics for data quality are no longer valid. Thus the same data set may be valuable for a particular usage, but useless for another.

Consider for example a virtual store that is compiling a comparative price list for a given product (such as Google products, previously known as froogle) through a meta search (a search that queries results of other search engines and selects best possible results amongst them). It obviously does not read all the millions of results for a search and does not return millions of records to the user. It normally selects top k results (where k is a constant value) from each search engine and finally returns top n results after the merge.

In the above scenario, when a user queries for a product, the virtual store searches through a variety of data sources for that item and ranks and returns the results. For example the user may query for "Canon PowerShot". In turn the virtual store may query camera vendor sites and return the results. The value that the user associates with the query result is clearly subjective and related to the user's intended requirements, which go beyond the entered query term, namely "Canon PowerShot" (currently returns 91,345 results from Google products). For example the user may be interested in comparing product prices, or the user may be interested in information on latest models.

More precisely, suppose that the various data sources can be accessed through a view consisting of columns ("Item Title", "Item Description", "Numbers Available", "Price", "Tax", "User Comments"). A user searching for "Canon PowerShot" may actually be interested to:

Learn about different items (products) - such a user may not care about the "Numbers Available" and "Tax" columns. "Price" is somewhat important to the user although obsoleteness and inaccuracy in price values can be tolerated. However, consistency of "Item Title" and completeness within the populations of "User Comments" in the query results, is of highest importance.

Compare prices - where user is sure about the item to purchase but is searching for the best price. Obviously "Price" and "Tax" fields have the greatest importance in this case. They should be current and accurate. "Numbers Available" is also important although slight inaccuracies in this column are acceptable as any number more than 1 will be sufficient.

Above examples indicate that selection of a good source for data is subjected to what does the term "good" mean to the user. In this research, we propose to include user specific quality considerations into query formulations, in order to address user specific requirements. We term this as quality-aware queries. Quality aware queries are a multi-faceted problem. Aggregations across multiple large data sets are infeasible due to the scale of data. Further, ranking approaches based on generic user feedback gives a constant rank to the quality of a data source and does not factor in user/application specific quality ranking.

This research investigates the exploitation of user specific data quality (DQ) criteria to respond to user queries in multi-data source systems. We discuss how how they can be captured from users, how DQ measurements can be acquired from datasets, and how the above can be used to improve the quality of query results with respect to fitness for use. The above questions are addressed within an overarching framework that provides the necessary tools and services for DQ aware query processing.

VLDB 2010 PhD Workshop, September 13, 2010, Singapore

The remaining report is organized as follows. Before we define our research challenges in Sec. 4, we briefly discuss related works in Sec. 2, and present the overall research framework and solution architecture in Sec 3. In Sec. 5 to 7 we discuss solutions to each of our challenges. Section 8 describes datasets and experiments used to evaluate our approaches, and we conclude in Sec. 9.

2. STATE OF THE ART

Consequents of poor quality of data have been experienced in almost all domains. Due to space limitations, we only highlight a few researches in this section. From the research perspective, data quality has been addressed in different contexts, including statistics, management science, and computer science. To understand the scope, various research works have defined a number of quality dimensions [9]. To address the problems that stem from the various data quality dimensions, the approaches can be broadly classified into investigative, preventative and corrective. Investigative approaches essentially provide the ability to assess the level of data quality and are generally provided through data profiling tools (See e.g. IBM Quality Stage). A variety of solutions have also been proposed for preventative and corrective aspects of data quality management. These solutions can be categorized into following broad groups: Semantic integrity constraints solutions [3]. Record linkage solutions. Record linkage has been addressed through approximate matching, de-duplicating and entity resolution techniques [1]. Data lineage or provenance solutions are classified as annotation and non-annotation based approaches where back tracing is suggested to address auditory or reliability problems. Data uncertainty and probabilistic databases are another important consideration in data quality. The data quality problem addressed in this work, also has relevance to preference theory [8]. [2] develops a logical framework for formulating preferences in databases. However, previous research does not consider the preferences on data quality. A part of our research deals with the query planning problem. Query planning is a wide area of study. A good survey of adaptive query planning can be found in [4]. A completeness based query planning method is studied in [7]. We are not aware of any query planning approach that considers the quality of data as well as user preferences on DQ into account.

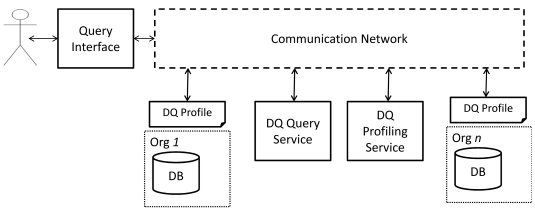

Figure 1: DQ aware query system architecture

3. FRAMEWORK

The challenges addressed in this research are positioned within an overall framework, namely Data Quality Aware Query System (DQAQS). A wrapper-mediator architecture is assumed for DQAQS as in Fig. 1. Each data source consists of a base (relational) database and certain meta data. The meta data consists of two main components: i) Schema and ii) Data Quality profiles. Each source is responsible for providing data quality profiles for each of its datasets. The DQ profiling service generates DQ profiles and calculates DQ metrics (which is the measurement for a quality attribute of data) for a given dataset. DQ aware query service is the query engine that conducts the user queries. Query interface uses DQ aware query service to run quality aware queries and return the results to the user.

4. RESEARCH TASKS

The general problem of DQ aware query processing, can be divided into three subtasks, and consequently into three research questions. In this section, we discuss these challenges focussing on their role in developing DQAQS.

Profiling Data Quality. Profiling is the task of collecting descriptive statistical information about data. These statistics can in turn be used in query planning, and query optimization. Data quality profile is a form of meta data which can be made to the query processing engine to predict the quality of the query result, and in the presence of data sources, to optimize the quality of query results.

Information collected in data quality profiles clearly affects the effectiveness of the query systems ability to process quality aware queries. Data quality profile can be generated with different granularities both horizontally and vertically. We call profiling per data source, per relation, and per attribute as vertical granularity of DQ profile. We call the per tuple, per selected subsets of dataset or per dataset as horizontal granularity. Obviously, richness of the profile will contribute to the effectiveness of the query making predictions on the quality of the source/result-set closer to reality but as always it comes with a trade-off with storage and performance.

In today's technology, data storage is rarely a problem, and user satisfaction with the query results can be deemed more important than storage. In this research, we propose methods and techniques for profiling data quality. The challenge is to create a near minimal DQ profile Pr of a dataset D, such that using Pr, quality of the result of any query against D can be estimated with a guaranteed degree of certainty.

User Preferences on Data Quality. Modelling user preferences is a challenging problem due to its inherent subjective nature. Additionally, DQ preferences have a hierarchical nature, since there can be a list of different metrics for each attribute in the query. Several models have been developed to model user preferences by decision making theory and database communities. Models which have been based on partial orders are shown to be effective in many cases [8]. Different extensions to the standard SQL have also been proposed to define a preference language [6].

Inconsistencies in preferences occur often as preferences are user defined. Current studies on user preferences in database systems assume that existence of inconsistency is natural (and hard to avoid) for user preferences and a preference model should be designed to function even when user preferences are inconsistent, hence; they deliberately opt to ignore it. Nevertheless, all studies do not always agree with this assumption [5]. Human science and decision making studies show that people struggle with an internal consistency check and they will almost always avoid inconsistent preferences if those individuals are given enough information about their state in their decision (e.g. visually). [5] believes

inconsistencies in user preferences are most likely the result of human mistakes caused by confusion. Here, the challenge is to propose of a method and formulate a language (preferably based on SQL standards) to model user preferences on data quality, as well as to provide a simple, fast and effective graphical user interface, to capture user preferences on DQ. This interface should help user in defining preferences by visually informing the user on possible inconsistencies, which involves consistency detection algorithms.

Data Quality Aware Query Planning. DQ aware query planning is different from the general query planning problems studied extensively in database research. A DQ aware query plan is the combination of projected attributes from relations of different data sources linked together in a way that satisfies the query requirement. For example if there are several data sources $S_1, .., S_n$ available for relation *Items(Brand ,Model ,Price)*, a possible query plan is $\pi_{Brand,Model}(R_{S_1}) \ltimes \pi_{Price}(RS_2)$ which defines that for the query specified by the user, attributes $Brand, Model$ can be queried from the source S_1 and attribute $Price$ can be queried from the source S_2. A DQ aware query planner should be able to predict the quality of the result set of the plan. In fact, a DQ aware query planner should search through the planning search space to find the plan that has the highest result-set quality. Query plans that do not return results should be discarded.

The problem of DQ aware query planning is that quality of the result of joining attributes from two different datasets can be completely different from the quality of original datasets. For example assume the completeness of attribute $Price$ in databases S_1 and S_2 are both 50%. If all incomplete tuples of database S_1 are tuples with the brand $Sony$ and all incomplete tuples of database S_2 are tuples with the brand $Cannon$, completeness of the price of the result set will be zero. Further, in the presence of large data sets, any proposed methods will need to make appropriate performance/efficiency considerations.

Another challenge is to develop a method to rank the query plans based on their quality and user preferences on data quality. For example if user specifies that the quality of "attribute $Price$ is highly preferred to attribute $Model$" in regards to data quality, plans that return result set with very high quality $Price$s should be ranked higher even if their quality of attribute $Model$ is low.

5. PROFILING DATA QUALITY

Data quality dimensions characterize data properties e.g. accuracy, currency, completeness, consistency, etc. Many dimensions are defined for assessment of quality of data that give us the means to measure the quality of data. Measurements made on a dataset for DQ dimensions are called DQ metrics and the act of generating DQ metrics for DQ dimensions is called DQ profiling. However, definition of a DQ dimension may vary between different organizations. We define a DQ metric function as a set of rules that describe the DQ dimension. We assume that a set of DQ metrics M is standardized between data sources, however data sources may have different approaches (i.e. different rules) to calculate their DQ metrics (e.g. an England based data source has a different set of rules from an Australian based data source for checking the accuracy of address). We propose methods to generate DQ profiles vertically for source level, relation level and attribute levels, and horizontally for data set level and sub sets of data set levels.

Given data source S, and a set of data quality metrics M, source level DQ profiling is to calculate the value m_S for each metric $m \in M$, where m_S determines the possibility of a given record from the source S be considered as good quality per description (rules) of the metric m. Similarly, relation level DQ profiling is to calculate m_R for relation $R \in S$, where m_R is the possibility of any given record from relation R be considered as good quality per description of the metric m. Likewise, m_T determines the possibility of a given tuple from relation R be considered to have good quality for the value of attribute $a \in R$.

Let $\{a_1, \ldots, a_m\}$ be all attributes of the relation R representing dataset $D \in S$, and metric m be a set of rules. We define metric function $m_a(t)$, $t \in \zeta, \zeta \subseteq D$ as 1, if the value of attribute a from tuple t, does not violate any rule in m, and 0 otherwise.

For example, consider m as Consistency metric, includes a rule that checks for functional dependency. If tuples t_1 and t_2 are functionally dependent, $m(t_1), m(t_2)$ both should return 0 (failed), but $m(t_3)$ may return 1 if it passes all the rules in m.

Attribute level profile m_a for dataset D, metric m and attribute a is defined as $\frac{\Sigma_{t \in D} m_a(t)}{|D|}$. Relation level profile m_R for dataset D, and metric m, is defined as $m_R = avg(m_a), a \in R$. Source level profile m_S for data source S, is not usually calculated from data, instead; it is based on user feedback.

Attribute level profile over the whole dataset does not provide enough information to predict DQ of the query result set. For example, a new car dealer may have used cars in its database also. Since they are more particular about data entry of their own cars, quality of the used-car subset of database is much less than the quality of the new cars subset of DB. The only situation where attribute level metric value of the whole dataset will be similar to the attribute level metric value of any subset of the dataset is when distribution of dirty data within data-set is evenly random. Data quality profile that stores attribute level DQ statistics for the whole dataset will require very little amount of storage for each data source, but a full scan of database for generation of DQ measurements is required, regardless.

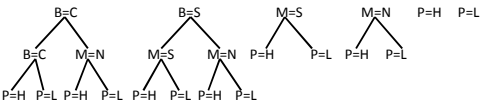

Figure 2: Search space for DQ profile generation

Development of DQ profiling methods to generate a minimal DQ profile such that quality of any projected subset of the original dataset can be estimated from it is a challenging problem which we call advanced DQ profiling.

We define the problem of advanced DQ profiling as follows: Given dataset D of relation R, attribute a, and certainty threshold ϵ find the minimum set of tuples P_R called DQ profile Pr of R that are enough to predict $m_a(\sigma_\Phi(D))$ for an arbitrary selection operation $\sigma_\Phi(D)$ with no more than ϵ incorrectly predicted tuples where Φ is a selection condition consisting of $\wedge$ and $\vee$ operators.

We assume that any attribute $a \in R$ has a limited domain of values. We define $dom(a)$ as limited domain of the values of the attribute a in addition to the special value "-" as

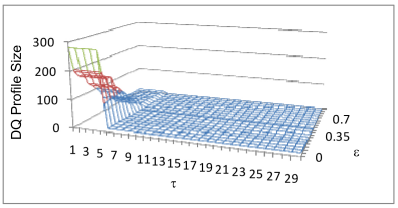

Figure 3: Effect of certainty threshold on the size of DQ profile

don't-care (which can sit instead of any value). For simplicity, we also assume that conditions within Φ are only consist of equality comparisons since in a finite domain, range comparisons can be defined as a set of equality comparisons.

If we use brute-force search the whole domain of possible selection conditions over dataset D, pre-compute the metric function for each possible condition and store results and the selection condition in DQ profile P, then we can later query P to exactly predict the quality of the result set for any given query with any selection condition. For this reason, the search space not only includes all possible equality comparison for attribute a (i.e. $\{\Phi = a\ equals\ d | d \in dom(a)\}$), but also it includes all possible $\wedge$ combination of the equality conditions. We observe that $m_a(\sigma_{\Phi_1 \vee \Phi_2}(D))$ can be directly calculated from $m_a(\sigma_{\Phi_1})$ and $m_a(\sigma_{\Phi_2})$, hence, profile data is independent from the $\vee$ operator.

Figure 2 depicts a sample search-space for brute-force method for sample metric m. Database for Fig. 2 consists of relation $Items(Brand, Model, Price)$. Each attribute is identified with it's first letter, i.e. B, M, P and their values are restricted to following domains: $dom(B) =$ { Sony, Cannon }, $dom(M)$={ SLR, Norm}, $dom(P)$={ Low, High}. Only first letter of the value is shown in the equality comparisons of Fig. 2. First two trees ($B = C, B = S$) browse the whole database, and rest of the trees are redundant. In fact, by having DQ metric values for the first two trees, metric function result of any query over relation $Items$ can be predicted. For example, to predict $m(\sigma_{P=H \wedge M=S}(Items))$ all nodes $B = S, M = S, P = H$ and $B = C, M = S, P = H$ should be traversed.

The brute-force search is exhaustive and extremely expensive, but studying it helps to understand the problem. Considering the fact that metric functions are probability functions, we compromise its exact prediction with considerable improvements in speed and disk space. Our study on some datasets shows that significant reduction in the size of the profiling dataset can be achieved with tolerating some degree of inaccuracy (in Figure 3 profile size is less than 5% of size of the database). Figure 3 illustrates the effect of the certainty threshold on the size of the minimized profile dataset. The original dataset used for Fig. 3 is about 4000 records and the metric function in use is completeness. Horizontal axis shows the maximum number incorrectly predicted records, the depth axis shows the smallest result-set that it's DQ metric can be predicted and the vertical axis shows size of the profiling dataset.

This is ongoing study and challenges remain to minimize size of the profiling dataset, with a guaranteed user defined upper band of uncertainty and indexing it for vary fast querying.

6. USER PREFERENCES ON DATA QUALITY

Intuitively preferences are regarded as sets of partial orders [8]. For example "I like coffee more than tea" or "I like juice more than coffee" are partial order clauses that can describe user preferences. To be more precise, a partial order clause can convey strength of a preference. For example "I like coffee much more than tea" and "I like juice slightly more than coffee". People express the strength of each preference with subjective adjectives such as very strongly, strongly, slightly, etc.

The notion of Hierarchy in preferences is defined in the literature [8] as prioritized composition of preferences. For example; considering relation $Item\{Brand, Model, Price\}$ completeness of the prices may have priority over completeness of models. We use the term Hierarchy to define prioritised composition of preferences which can form several levels of priority or hierarchy. The hierarchy or priority over the preference relations is quantifiable such as: a is strongly more important than b, or a is moderately more important than b.

We divide approaches for capturing user preference on DQ into two sections. First, in [11] we proposed a preference language and an extension to SQL for modelling user preferences on DQ. Then, we develop methods to detect and repair inconsistencies in user preferences on DQ.

Data Quality Preferences Language Let relation $R(A)$ be a relation of attributes $a \in A$. Let $M = m_1..m_k$ be the set of k data quality metrics. Let $S = S_1...S_n$ be a set of possible sources for relation $R(A)$. A preference formula (pf) $C(S_1, S_2)$ is a first order formula defining a preference relation denoted as $\succ$, namely

$$S_1 \succ S_2 iff C(S_1, S2).$$

Let relation $R(A)$ be a relation of attributes $a \in A$. Let $M = m_1..m_k$ be the set of k data quality metrics. Let $S = S_1...S_n$ be a set of possible sources for relation $R(A)$. A preference formula (pf) $C(S_1, S_2)$ is a first order formula defining a preference relation denoted as $\succ$, namely

$$S_1 \succ S_2 iff C(S_1, S2).$$

A hierarchy (prioritized order) of preferences is defined as: Consider two preference relations $\succ_1$ and $\succ_2$ defined over the same schema. The prioritized composition $\succ_{1,2} := \succ_1 \triangleright \succ_2$ of $\succ_1$ and $\succ_2$ is defined as:

$$S_1 \succ_{1,2} S_2 \equiv S_1 \succ_1 S_2 \vee (\neg S_2 \succ_1 S_1 \wedge S_1 \succ_2 S_2).$$

Inconsistency Detection Due to hierarchical nature of preferences, the uncertainty that happens as a result of inconsistency in user preferences is noticeable since uncertainties propagate to lower levels of preference hierarchy, thus eventually compromising query response. Hence, methods are needed to identify inconsistencies, as well as to notify user about it. Even though inconsistency in the user preferences could be accepted sometimes, informing the user of inconsistencies has no negative effects.

A preference query consists of a set of prioritized orders $\succ_{x,y} w$ where w is the weight of the priority and x and y are other preferences which can be recursively prioritized preference. Inconsistency detection problem can thus be defined as: Given a prioritized preference $\succ_{x,y} w$ within the preference query, any other recursively inferred prioritized

preference should be same as $\succ_{x,y} w$. Searching for inconsistent set of pair prioritized preferences is not trivial.

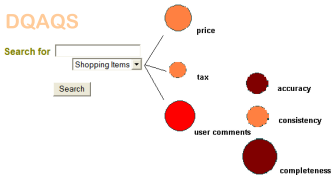

Figure 4: Configuring preferences for a DQ aware query

We modelled preferences as directed weighted graphs that we efficiently search for inconsistencies using a heuristic we developed in [10]. However proposition of minimal changes to the query to fix the consistency problem still need to be addressed. In addition, we have developed an interactive graphical user interface for DQAQS to effectively capture user data quality preferences. Figure 4 shows a screenshot of this user interface for a simple query from "Shopping Items". A natural hierarchy of the query attributes and their quality metrics is represented as a tree of connected circles. Size of a circle compared to other circles identifies its priority and colors are designed to imply consistency of DQ preferences to the user.

7. DATA QUALITY AWARE QUERY PLANNING

Query planning is a classical combination-optimization problem. Solutions to such problems can be described as states in a space of semantically equivalent states. A query planner starts at an initial state and manipulates it in such a way that the optimal or at-least near optimal state is reached while the optimization goal (i.e. highest quality) is satisfied. DQ aware query planning can become fairly complex since the user preference on DQ metrics need to be considered as a key factor in the optimization goal. We define the problem of DQ aware query planning as follows:

Let S_i, $1 \leq i \leq n$ be the all data sources, and $\Im$ be the global mediated schema, that for any relation R, defines data sources $\{S_i\}$ where $R \subseteq S_i \in \Im$. Let P_i be the profiling data set for data source S_i and M be the set of data quality metric functions.

Given query $Q = \sigma_{\Phi_1}\pi_{\alpha_1}(R_1) \ltimes \ldots \ltimes \sigma_{\Phi_n}\pi_{\alpha_n}(R_n)$, $\alpha_i \subseteq R_i$ and a hierarchy of prioritized DQ preferences consisting of $\succ_{a_j,a'_j}, a_j \in \alpha = \{\alpha_1 \bigcup \ldots \bigcup \alpha_n\}$, and $\succ_{m_j,m'_j}$, $m, m' \in M$, for each $a_j \in \alpha$, find top k DQ aware plans as $Pl = \pi_{\alpha'_1}(S_1) \ltimes \ldots \ltimes \pi_{\alpha'_{n'}}(S'_n)$. Plan Pl defines that set of attributes α'_i should be selected from data source S_i. The only constraint is that α should be the same as $\alpha' = \{\alpha'_1 \bigcup \ldots \bigcup \alpha'_{n'}\}$.

We consider a three part strategy to rank top plans for the given query Q. Our strategy involves answering three questions: First, How to estimate the quality of a plan? Second, how to weight query items or how to identify the real importance of each attributes and its DQ metrics using prioritized preferences $\{\succ_{m_j,m'_j}\}$? And third, how to search through the planning space for plans that satisfy the user DQ requirements the most.

Estimate the quality of plans. To address the first question, for query plan $\sigma_{\Phi_0}\pi_{a_0}(S_i)$, quality of the plan can be estimated by decomposing condition Φ_0 into equality conditions and operators. We refine relational algebraic operators to work with DQ profiles to estimate the quality of a subset of a dataset. Figure 5 illustrates a simplified DQ profile data set for relation $Items(Brand, Model, Price)$, and metric $m_a(R_{S_1})$ where a is an arbitrary attribute from relation R from the data source S_1.

Data Source: *S*, Relation: *Items*

Attribute: *Price* & Metric: *m*				
Brand	Model	Price	QC	Cnt
Sony	SLR	-	55	60
Sony		Low	12	35
Sony	-	-	60	90
Cannon	SLR	-	170	200
-	-	-	297	385

Figure 5: A sample DQ profile for data source S, and relation $Items$

For query $Q'' = \sigma_{Model=SLR \wedge Price=Low}(R_{S_1})$, $m(Q'')$ cannot be directly inferred from the DQ profile dataset. This happens when the number of tuples returned by the query is less than the accuracy threshold ϵ (as described in 6) or the quality of the result set can be statistically inferred from its closest superset, which in this example is $\sigma_{Model=SLR}$. Hence, $m(Q'') = (55 + 170)/(60 + 200) = 0.86$. The actual quality of the result set is between $0.86 - (\epsilon/|Q''|)$ and $0.86 + (\epsilon/|Q''|)$ since DQ profile dataset is optimised such that accuracy threshold ϵu is guaranteed.

Estimation of the quality of a query plan that consists of a join between different relations is highly dependent on the quality of the join's key attribute. Since DQ profiles convey information about the domain of attributes, quality of joins can be estimated by calculating the intersection between domains. It is not always feasible to guaranty the estimation results for complex joins, when the sub-set of DQ profile after selection operators on different attributes, does not have enough information about the sub-domain of the key attribute in the result-set. A possible solution to this limitation can be to generate a set of candidate sources for joins, and calculate the value of DQ metric function on the actual result set after querying data sources.

Weight query items. We employed Analytical Hierarchy Process (AHP) technique to address the second question. AHP technique is a decision making technique that at first, forms a hierarchy of items within the problem. Then for each level of the hierarchy, a priority matrix is generated that reflects all mutual weighted priorities. If the hierarchy and all the priority matrices are provided, AHP technique can weight all items in the query with a number. To generate priority matrices, if a hierarchy a over a' is not defined in the user query, we estimate it as follows: If there is no priority a' over a with weight t, use a priority of a over a' with weight 1, otherwise use a priority of a over a' with weight $1/t$.

For example given a query with DQ preferences such as $Price$ is highly preferred to $Model$ and $Model$ is preferred to $Brand$, and metric m_1 is slightly preferred to metric m_2, approach described above assigns a numeric weights such that numbers 1 to 9 represent subjective weights from low to high [8]. w_j, w_ij to each attribute a_j or metric m_{ij}, where $m_i \in M$ and m_{ij} defines metric m_i for attribute a_j.

Search for top plans. To address the third question, if the query plan consists of a single data source, sources can be ranked using a number of ranking methods [7]: Simple Ad-

ditive Weighting (SAW), etc. In [7] a comparative analysis of the mentioned ranking methods is provided which shows that the effectiveness of all above methods is not considerably different with regards to the source selection problem. The major difference of these methods is their computational complexity. So far we have experimented with the SAW method which is easy and fast. The SAW method involves three basic steps: Scale the scores to make them comparable, apply the weighting, and sum up the scores for each source. Using DQ profiles, we calculate metric function $m_{j,a_i}(S)$ for metric m_j and attribute a_i from data source S. Data sources can be ranked using the quality score $dq(S) = \sum w_i M_i$, where $dq(S)$ is the final weight of source S, w_i is the weight of attribute a_i which has been calculated through the AHP process and $M_i := \sum w_{ij} m_{j,a_i}(S)$.

As described earlier, in some join queries, guaranteed estimation of the quality of the plan may not be possible. In such scenarios, SAW method can not be used since a metric function for a plan can not be calculated. Further challenge is to develop a method to search within the plan space of complex inter-data source joins. One method is to candidate a set of data sources that are more probable to play in the top plans. Then calculate the actual quality of the query results instead of estimating them. Selection of such candidates is challenging since the candidate set should be effective, but minimal.

8. EXPERIMENTS

We acknowledge the need to evaluate the proposed measures, heuristics and methods against distributed, large, realistic, and interesting data sets. Various modules of our architecture for DQAQS as describes in Sec. 3 serve this purpose and are under development. So far we have developed a DQ profiling service, which is implemented using web service technology; profiling service has the possibility to mirror itself and required parts of the rules dataset as a local function when required, in order to overcome technical limitation of slow web services in time-critical applications. We have also developed a SQL based query interface as well as a web-based graphical user interface as query facilities for the user. In order to conduct experiments in DQAQS, we utilize synthetic as well as real world data. We use syntactic data to control several parameters, such as number of data sources, type and distribution of errors. We have developed a tool to generate duplicate datasets with different qualities and data domains from a single data source. Real world scenarios considered so far include a time-series dataset from a power plant as well as freely available data sets such as DBLP.

9. CONCLUSION

Our research in data quality aware query systems, addresses three major challenges. First, we investigate on how meaningful statistical representation of databases, namely DQ profiles can be generated in a way that quality of any query can be effectively predicted from it, without actually querying data sources. The generated DQ profile should be markedly small compared to the original data set, and estimation of the quality of query results from them needs to meet efficiency requirements. Second, we are studying and developing a DQ preference model to capture user requirements on data quality, assuming a hierarchical structure of DQ queries, where any attribute in the query can have several data quality metrics linked to it. As fallout of capturing user preferences, we propose methods to detect inconsistency in user preferences and also design intuitive graphical user interfaces to effectively interact with non technical users for their data quality preference specifications. Third, we investigate DQ aware query planning techniques that involves estimation of the data quality of data sources, and ranking of plans to maximise user satisfaction. The proposed query engine considers user preferences on data quality when it searches for query plans. We hope that the pursuit of problems in this research will lead to DQ aware query systems, and our insights and solutions will contribute to higher user satisfaction and awareness of data quality for a wide range of applications.

10. ACKNOWLEDGEMENTS

This research is supported by the Australian Research Council (ARC grant no. DP0773122). Besides my supervisor Associate Professor Shazia Sadiq, I would also like to acknowledge the helpful discussions with Prof. Xiaofang Zhou, Dr. Ke Deng, and Dr. Henning Koehler that contributed to the development of ideas in this research.

11. REFERENCES

[1] O. Benjelloun, H. Garcia-Molina, Q. Su, and J. Widom. Swoosh: A generic approach to entity resolution. *VLDB Journal*, 2008.

[2] J. Chomicki. Querying with Intrinsic Preferences. *Lecture notes in computer science*, pages 34–51, 2002.

[3] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma. Improving data quality: consistency and accuracy. *Proceedings of the 33rd international conference on Very large data bases*, pages 315–326, 2007.

[4] J. Hellerstein, M. Franklin, S. Chandrasekaran, A. Deshpande, K. Hildrum, S. Madden, V. Raman, and M. Shah. Adaptive query processing: Technology in evolution. *Bulletin of the Technical Committee on*, page 7, 2000.

[5] J. HEY. Do Rational People Make Mistakes? *Game Theory, Experience, Rationality: Foundations of Social Sciences, Economics and Ethics: in Honor of John C. Harsanyi*, page 55, 1998.

[6] W. Kießling. Foundations of preferences in database systems. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 311–322. VLDB Endowment, 2002.

[7] F. Naumann. Quality-Driven Query Answering for Integrated Information Systems. LNCS 2261, 2002.

[8] T. Saaty. *Multicriteria Decision Making: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. RWS Publications, 1996.

[9] R. Wang and D. Strong. Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–33, 1996.

[10] N. Yeganeh and S. Sadiq. Avoiding Inconsistency in User Preferences for Data Quality Aware Queries. *Submitted to Business Information Systems*, 2010.

[11] N. Yeganeh, S. Sadiq, K. Deng, and X. Zhou. Data Quality Aware Queries in Collaborative Information Systems. *Advances in Data and Web Management*, pages 39–50, 2009.