Completeness of Information Sources

Felix Naumann
IBM Almaden Research Center
felix@almaden.ibm.com

Johann Christoph Freytag, Ulf Leser
Humboldt-Universität zu Berlin
freytag@dbis.informatik.hu-berlin.de
leser@informatik.hu-berlin.de

***Abstract*— Information quality plays a crucial role in every application that integrates data from autonomous sources. However, information quality is hard to measure and complex to consider for the tasks of information integration, even if the integrating sources cooperate. We present a systematic and formal approach to the measurement of information quality and the combination of such measurements for information integration. Our approach is based on a value model that incorporates both extensional value (coverage) and intensional value (density) of information. For both aspects we provide *merge functions* for adequately scoring integrated results. Also, we combine the two criteria to an overall *completeness* criterion that formalizes the intuitive notion of completeness of query results. This completeness measure is a valuable tool to assess source size and to predict result sizes of queries in integrated information systems. We propose this measure as an important step towards the usage of information quality for source selection, query planning, query optimization, and quality feedback to users.**

I. Introduction

With the increasing interconnection of information systems, the integration of information from various autonomous and independent data sources is becoming a more and more important topic. Many free applications can be found in the Web, such as meta search engines, integrated stock information systems, or bibliographic services. Commercial applications include typical eBusiness applications such as marketplaces and eProcurement systems—both relying on catalog integration—and inter- or intra organizational projects for enterprise application integration (EAI). A recent announcement of SAP, stating that future versions of SAP R/3 application will be entirely based on an *information integration layer* supports this claim [1]. No matter how the integrated data is stored, if a virtual or a materialized integration approach is followed, and what data models and schemas are used, *information quality* plays a crucial role. This emphasis is especially true for independent and autonomous sources. Thus, the paradigm of information querying is dramatically changed: Because users assume a centralized database management system to always have all the information, i.e., to provide a *complete* result inherently, the merit of such a system is measured by its speed or response time. But the assumption of completeness no longer applies to integrated information sources. The response time of an Internet source is less crucial compared to its ability to provide the information queried for. For example, most freely available stock information systems do not provide data about all world-wide stocks. Also, these systems do not provide all the information there is for a certain stock.

To gain the full advantage of multiple sources, a user must query all available sources and integrate the results. Even if automated, this is a tedious and often expensive task. Some measure is needed to determine which sources are to be preferred over others. This measure must take into account both the number of objects provided by the source, and the amount of information per object it provides. For the example of stock information systems, these numbers include the number of stock quotes covered by the source and the number of attributes per stock quote it provides (such as current score, days range, company profile etc.). In this paper, we describe a complete framework for dealing with the specification and integration of the quality of information provided by a single data source or by a set of data sources. We consider the intensional character of information quality (called density) and the extensional character of information quality (called coverage). In essence, the coverage measure describes how many objects an information source can provide; the density measure describes how much data for each of those objects the source can provide. When data from sources are merged, the scores for coverage and density of the merged result must be estimated. For both quality dimensions we therefore provide *merge functions*. Finally, we combine the two aspects to an overall *completeness* criterion. This completeness measure is a handy and valuable tool to assess the quality of data sources and of combinations of data sources, which leads to various applications such as source selection, query optimization, bounding result sizes, etc.

a) Related work: Some other projects have strived to model the "size" of information sources. Chen et al. mention the criteria "Size of result" and "Number of documents accessed" but neither define them, nor point out the difference of the two, nor show how to integrate the two into a general value model [2]. Motro and Rakov define a "completeness" criterion [3]. The criterion matches our coverage criterion. However, their research does not go beyond the mere definition, whereas we enhance the model by defining the coverage of combinations of sources. Also, we combine coverage with the density criterion and only then capture the true value of information sources. To the best of our knowledge the density criterion as we define it has never before been addressed in literature. Florescu et al. quantitatively describe the content of distributed autonomous document sources using probabilistic measures [4]. In their model, the authors calculate two values: "Coverage" of data sources, determining the probability that a given document is found in the source, and "overlap" between two data sources, determining the probability that an arbitrary document is found in both sources. These probabilities are calculated with the help of word-count statistics. Their coverage measure is similar to the *precision* measure of the information retrieval field and determines the query dependent usefulness of a source. Their overlap measure expresses ideas similar to ours, but the authors do not consider different types of overlap, such as independence or disjointness.

b) Structure of this paper: What follows is a general definition of data merging operators. On top of these operators, we provide a thorough definition and analysis of the two criteria *coverage* and *density*. For each criterion we show how to merge scores of different sources with varying overlap situations. Finally, we combine the two criteria into the general *completeness* criterion. A "stock information system" example guides the reader through our approach.

II. Querying Cooperative Information Systems

This section gives a general introduction to our setting with special attention to our application example of an integrated stock information service. We describe the type of information we query, the ways of accessing the information at multiple sources, and how results from the sources are merged to present the query response to the user.

A. The information model

- Global schema: We assume a global schema that consists of only one relation. This relation contains a globally unique ID and the union of all attributes delivered by sources. A *user query* is a selection of different attributes. A source is described as a view on the global schema that projects out any attribute not delivered by the source. Each source must deliver the ID attribute. We assume heterogeneity to be resolved elsewhere, e.g. through specialized data wrappers (see below).

  Having only one global relation seems overly restrictive, but is in many cases a convenient and sufficient model (see work on the *universal relation* by Maier et al. [5]).
- Globally consistent IDs: We assume that each object has a globally unique identifier that is stored at the sources. The identifier is consistent across sources, i.e., if two sources present an object with the same ID, then the we consider these objects to represent the same real-world entity. IDs are merely used to merge information; we do not require that they define functional dependencies, nor that each source stores at most one object per ID. IDs are called "merge attributes" in [6].

  Although this assumption may seem overly strong, it is true for many domains: Stocks have their symbol as a global ID, books have an ISBN, persons have a passport number etc. If no such ID is available, we assume that one can be constructed. For instance, if complete information is present, a person ID could be the combined name and address fields (see [7] for further techniques).
- Overlap: We assume that source contents overlap to various degrees with each other regarding the objects they store information about. In an extreme case one source can be a mirror of another source, i.e., they totally overlap[1]. Other degrees of overlap are
  - *Containment*: The IDs in one source are a subset of the IDs in another source; the contained source stores only information about objects that the other source also stores information about. Nevertheless, the actual information (the attribute values) might differ.
  - *Independence*: There is no (known) dependency between the IDs of two sources. This means that there is some coincidental overlap, which we estimate using the size of the sources and the size of the real world they model. Whenever there is no knowledge about containment or disjointness, we assume independence.
  - *Disjointness*: The sources provide no ID in common.

  In general, querying several sources with little overlap retrieves a larger number of distinct objects with only some attribute values. Querying several sources with large overlap retrieves a smaller number of objects but likely with more attribute values.

Example. We use a meta stock information service (MSIS) as an example to guide intuition to our completeness approach. An MSIS is a system that provides information on stock quotes. Unlike ordinary stock information systems (SIS), the MSIS combines information from several systems. A search request is sent to a whole set of SISs, the results are merged and presented to the user in a homogeneous way. The results of SISs and MSISs alike are typically lists of stock symbols, their current quotes, and some additional information like the trade volume or the quote change. A query for *IBM* on a typical SIS may have the result shown in Figure 1.

This SIS delivers the symbol, time, last quote, change of quote, change percentage, and volume[2]. We ignore the links to more information. Other SIS may provide other information. To capture it all, we use the union of all attributes of all sources as global schema. In our examples, we consider the following 7 SIS:

- Yahoo finance (`finance.yahoo.com`)
- Yahoo Finanzen (German version, `finanzen.de.yahoo.com`)
- CNN stock quote service (`qs.cnnfn.com`)
- New York Stock Exchange (`www.nyse.com`)
- e*trade (`www.etrade.com`)
- AltaVista Money section (`money.altavista.com`)
- Merrill Lynch (`www.ml.com`)

Whenever an attribute is not provided by a source, the corresponding field is left empty (`null`-value) for all objects. Our global relation for SIS results is shown in Table I. The global IDs of SIS results are the stock *symbols*. The name

[1] Mirroring implies complete intensional and extensional equality. Our overlap definition applies only to extensional overlap.

[2] We used the Yahoo finance system for this example. The detailed table provides many more fields.

Mon Jan 17 10:29am ET - U.S. Markets Closed for Martin Luther King, Jr. Day.

Symbol	Last Trade		Change		Volume	More Info
IBM	Jan 14	119 5/8	+ 1 3/8	+1.16%	10,956,000	Chart, News, SEC, Profile

Fig. 1. An SIS query result

symbol	name	last trade date	l.tr. quote	change	change %	volume	td's high	td's low
...	...	...	...	...	...	...	...	...

TABLE I

GLOBAL RELATION FOR THE STOCK INFORMATION EXAMPLE.

attribute refers to the actual company name. The last trade date and quote are provided by all SIS. The two attributes are the most important information for typical users. The other attributes provide additional and statistical information and are available only in some of the 7 SIS. Some SIS provide much more information such as company profiles, charts etc. For simplicity we ignore those attribute in this report.

The results of SIS queries typically overlap; two systems may return information for the same stock or symbol. However, the set of attributes provided by the different SIS may differ. Also, the values of the attributes may differ from system to system, causing data conflicts in the result. These conflicts must be resolved by so called resolution functions. □

B. Result merging

An cooperative information system (CIS) distributes a user query to multiple information systems. After receiving the individual results, it is the task of the CIS to compile the results to a common response to the user. We call this process *result merging*. The merged result should be as consistent as possible despite conflicting data, and as complete as possible, i.e., contain all retrieved information. In general, a result merged from multiple sources contains objects where

1) some attribute value is not provided at all,
2) some attribute value is provided by exactly one source,
3) some attribute value is provided by more than one source.

Result merging of the CIS in the first case is clear – the object in the result has no value. How to merge information in the second case is also clear – when constructing the result, the one attribute value is used for the result object[3]. The third case demands special attention. Several sources compete in filling the result object with an attribute value. If all sources provide the same value, that value is used in the result. If this is not the case, there is a data conflict and some function must determine what value appears in the result table.

Definition 1 (Resolution function): Let D be an attribute domain and $D^+ = D \cup \perp$, where $\perp$ represents the `null`-value. A resolution function f is an associative function $f : D^+ \times D^+ \rightarrow D^+$ with

$$f(x,y) := \begin{cases} \perp & \text{if } x = \perp \text{ and } y = \perp \\ x & \text{if } y = \perp \text{ and } x \neq \perp \\ y & \text{if } x = \perp \text{ and } y \neq \perp \\ g(x,y) & \text{else} \end{cases}$$

where $x, y \in D^+$ and $g : D \times D \rightarrow D$. Function g is the internal associative resolution function that is responsible for resolving conflicting data. □

The generalization of Definition 1 to more than two input values is trivial. Resolution functions can be of various types, depending on the type of attribute, the usage of the value, and many other aspects as discussed by Yu and Meng [8][4]. One simple resolution function for strings might concatenate the values and annotate them with the source that provided the value. A resolution function for numerical values might be to determine the average value. To formalize result merging of entire query results and not single attributes, we define two new relational operators, the join-merge-operator denoted $\sqcap$ and the union-merge-operator denoted $\sqcup$. Both operators include resolution functions in case of data conflicts. We call both operators *merge* operators, because multiple results are merged to a common result. They are not simply concatenated, but objects appear only once in the result, possibly with attribute values from multiple sources. Missing values are padded with nulls. First, we define the join-merge-operator and show an example in Figure 2.

Definition 2 (Join-Merge, $\sqcap$): Let $R = (A_1, \ldots, A_m)$ and $S = (A_1, A_i, \ldots, A_n)$ be two relations with a common ID attribute A_1. The attributes $A_i, \ldots, A_m$ are common in both relations; they are each mapped to the same attribute in the global schema. Then

$$\begin{aligned} R \sqcap S := \{ & \text{tuple } t \mid \exists r \in R,\ s \in S \text{ with} \\ & t[A_1] = r[A_1] = s[A_1], & (1) \\ & t[A_j] = r[A_j],\ j = 2, \ldots, i-1 & (2) \\ & t[A_j] = f_{j-i}(r[A_j], s[A_j]),\ j = i, ..., m & (3) \\ & t[A_j] = s[A_j],\ j = m+1, \ldots, n \} & (4) \end{aligned}$$

where (1) is the join condition, (2) are the values provided only by *R*, (3) are the potentially conflicting values, and

[3] We assume $\perp$ values as missing knowledge.

[4] Information quality metadata can greatly enhance resolution functions, for instance favoring the more recent value.

(4) are the values provided only by S; and where $f_i(), i = 0 \dots m-i$ are attribute-specific resolution functions as defined in Definition 1. □

$r:$	A_1	A_2	A_3
	1	2	$\perp$
	2	5	$\perp$
	3	$\perp$	z

$s:$	A_1	A_3	A_4
	1	x	g
	3	y	$\perp$
	4	x	i

$r \sqcap s:$	A_1	A_2	A_3	A_4
	1	2	x	g
	3	$\perp$	$f_0(z,y)$	$\perp$

Fig. 2. The Join-Merge-operator

The union-merge operator is an extension of the join-merge operator. The union-merge-operator guarantees that every tuple from any source enters a join. An example is shown in Figure 3.

Definition 3 (Union-Merge, $\sqcup$): Let $R = (A_1, \dots, A_m)$ and $S = (A_1, A_i, \dots, A_n)$ be two relations with a common ID attribute A_1 and common attributes $A_i, \dots, A_m$. Then

$$\begin{aligned} R \sqcup S := {} & (R \sqcap S) \\ & \cup (R \setminus (R \sqcap S)[R] \times \{(\perp_{m+1}, \dots, \perp_n)\}) \\ & \cup (S \setminus (R \sqcap S)[S] \times \{(\perp_2, \dots, \perp_{i-1})\}) \end{aligned}$$

□

$r:$	A_1	A_2	A_3
	1	2	$\perp$
	2	5	$\perp$
	3	$\perp$	z

$s:$	A_1	A_3	A_4
	1	x	g
	3	y	$\perp$
	4	x	i

$r \sqcup s:$	A_1	A_2	A_3	A_4
	1	2	x	g
	2	5	$\perp$	$\perp$
	3	$\perp$	$f_0(z,y)$	$\perp$
	4	$\perp$	x	i

Fig. 3. The Union-Merge-operator

The union-merge operator is in nature similar to the full outer-join operator [9], but differs in one crucial aspect: The outer join does not allow the merging of columns from separate input relations into a single output column. Therefore, it does not deal with the issue of resolving conflicts and presenting a merged view of multiple sources. LaCroix and Pirotte defined a similar operator, the "generalized natural join operator", denoted $\overset{+}{\bowtie}$ [10]. Our merge operator differs from their approach in two aspects: First, data conflicts are resolved with a resolution function f. Second, our join is not a natural join; rather, the join predicate contains only one join attribute, the global ID.

Example. Imagine two stock information services delivering some results to a query for "IBM". Each returns a set of results, reflecting different IBM stock types traded at different markets. Some of the results might be common to both result sets, however with differing attributes and different attribute values. Some other results might be distinct to one of the result sets. To not lose any information, all results are merged with the union-merge operator in the result table. Figure 4 shows the two search results and the merged result for the user.

Observe that the first line of the merged result is not missing any attribute value. The two original sources complement each other in the information they provide, and combined they provide richer information. Wherever they overlap, some resolution function decides which value to choose. For instance, this is the case for the trade volume of IBM on that day. Because CNN states a higher volume, we must assume that that value is the more recent information and we choose it; an appropriate resolution function for this attribute is MAX(). This insight about the recency of a value could be used to decide upon conflicts among other attributes, such as ltq. A simple extension to our current definition of resolution functions would allow input other than the conflicting values. □

The following sections describe a measure to quantify the results of the join merge and union merge operators. The measure considers the number of results (coverage) and the number of attribute values in the result (density).

III. Coverage

We define *coverage* of a source to reflect the number of objects that a source can potentially return, i.e., the percentage of the real world the source *covers*. In this sense, coverage can be regarded as the *size* of a source. Coverage of a set of sources is the number of distinct objects that the set as a whole can potentially return. Because sources overlap to different degrees, it is a challenge to calculate the coverage of that set. The following sections discuss this matter. There is a strong relationship between coverage calculation and set theory. Sources can be viewed as sets of objects of the real world. The main difficulties of coverage calculation lie in determining the intersections of combinations of sources. Here, set theory can guide intuition and is used for proving several results.

A. Coverage of a source.

We define the coverage of a source as the ratio of the size of the source (number of distinct objects in the source) and the size of the *world*:

Definition 4 (The World): Given a global relation R of an application domain, we define W, called the *world*, as the set of all possible ID values of R that pertain to a real world object of the class modelled through R. The number of real world objects of R is $|W|$. Note that the actual value of $|W|$ and the values in W are irrelevant for all further computations. Only the size of W must be greater than the size of any source. □

Definition 5 (Coverage): Let S be a source or some other set of objects and let W be the set of real world objects. We

Yahoo finance:

symbol	name	ltd	ltq	change	change %	volume	td's high	td's low
IBM	$\perp$	10:45 AM	112 1/8	$+9/16$	+0.50%	1,458,600	$\perp$	$\perp$
IBM SICO.	$\perp$	9:47 AM	111	$+8/16$	+1.2%	677	$\perp$	$\perp$

CNN:

symbol	name	ltd	ltq	change	change %	volume	td's high	td's low
IBM	Intl. Business Machines	$\perp$	111 9/16	$-1/16$	$\perp$	1,529,500	112 13/16	111

Merged result (Yahoo $\sqcup$ CNN):

symbol	name	ltd	ltq	change	change %	volume	td's high	td's low
IBM	Intl. Bus. Mach.	10:45 AM	111 9/16	$+9/16$	+0.50%	1,529,500	112 13/16	111
IBM SICO.	$\perp$	9:47 AM	111	$+8/16$	+1.2%	677	$\perp$	$\perp$

Fig. 4. Two results for the query "IBM" (from Yahoo finance and CNN) and the merged response

define the *coverage* of a source as

$$c(S) := \frac{|S|}{|W|}.$$

□

Coverage is in $[0,1]$ and can be regarded as the probability that any given object of the real world is represented by some object in the source. In the following, if not specified otherwise, the coverage of a set of sources implies the coverage of the merge-union of these sources.

Example. About 40,000 companies are listed at stock exchanges all over the world, i.e., $|W| = 40,000$. Currently 3,114 of these are listed at the New York Stock Exchange and their quotes are available through their WWW information system. Other stock information systems combine stock quotes from several exchanges and thus gain a higher coverage. Table II shows the number of stocks listed at the individual systems together with their respective coverage scores. The coverage scores are obtained by dividing the number of stocks listed by 40,000. □

	Number	**Coverage**
Stock information system	**stocks listed**	**score**
Yahoo finance	10,095	0.252
Yahoo Finanzen	3,571	0.089
CNN stock quote service	9,375	0.234
New York Stock Exchange	3,114	0.078
e*trade	11,401	0.285
AltaVista Money section	12,000	0.300
Merrill Lynch	2,500	0.063

TABLE II

STOCK INFORMATION SYSTEM COVERAGE

B. *Coverage and overlap assessment*

The coverage measure for sources and sets of sources is based on timely and accurate coverage scores for individual sources. These scores are sometimes not easy to obtain. Often the sources themselves publish coverage scores as a means for advertising their service. However, not always can these figures be trusted. Another possibility to obtain coverage values is to simply measure coverage, where possible. Such assessment may be possible by downloading the source or querying the source. If these assessment methods fail, coverage scores can be estimated only by a domain expert. Overlap assessment is even more difficult. Equality, subset, or disjointness relationships can often be specified easily. But if none of the cases apply, the actual overlap should be determined. If this is not possible, one can assume independence[5]. Overlap information can be stored in a matrix, for which consistency can be checked.

Example. Overlap of two SIS is the number of companies that are listed in both services. With SIS it is often the case that one SIS is contained in another. For instance, Yahoo finance covers several SIS, such as the New York Stock Exchange SIS and the London Stock Exchange SIS. I.e., Yahoo finance is in itself a meta SIS, just like the one we propose with this example. Meta SIS can integrate other meta SIS and thus greatly enhance the service (and save much work). □

C. *Coverage of a set of sources.*

To respond to a user query in the best possible way, a query must be translated and submitted to multiple information sources. The results returned by these sources are sets of objects of the real world. Some objects may be returned by only one source but other objects may be returned by more than one source. To calculate the coverage of the merged result we must take into account the overlap between the different participating sources. What follows is a collection of intermediate results and the main result in Theorem 1. For brevity we omit all proofs and refer to [11]. In particular, we show how to calculate the coverage of the following terms, where S, S_i, and S_j are individual sources and P is a set of already merged sources:

- $c(S_i \sqcup S_j)$ for different overlap cases (Lemma 1)
- $c(P \sqcup S)$ for different overlap cases (Corollary 1)
- $c(S_i \sqcap S_j)$ for different overlap cases (Lemma 2)
- $c(P \sqcap S)$ for the general case (Lemma 3)
- $c(P \sqcup S)$ for the general case(Theorem 1)

Lemma 1 and its Corollary 1 motivate the different overlap situations and the proof of the Theorem 1. Lemma 2 and Lemma 3 show how to calculate parts of the result of the theorem. Finally, Theorem 1 covers the general case, where different kinds of overlap situations can occur simultaneously. The section is concluded by an example calculation of the coverage of a set of three search engines.

Lemma 1 ($c(S_i \sqcup S_j)$): Let S_i and S_j be the two sources to be union-merged ($S_i \sqcup S_j$). We distinguish the following cases:

[5] We assume independence if none of the other cases apply. Future research will deal with quantified overlap situations.

1) S_i and S_j are disjoint
$\Rightarrow c(S_i \sqcup S_j) = c(S_i) + c(S_j)$
2) S_i and S_j are independent
$\Rightarrow c(S_i \sqcup S_j) = c(S_i) + c(S_j) - c(S_i) \cdot c(S_j)$
3) $S_i \subseteq S_j$
$\Rightarrow c(S_i \sqcup S_j) = c(S_j)$

□

Once we compute the coverage of the merged result $S_i \sqcup S_j$, we can estimate the number of objects in $S_i \sqcup S_j$ as $c(S_i \sqcup S_j) \cdot W$.

Corollary 1 ($c(P \sqcup S)$): Let $P = \{S_1, \ldots, S_k\}$ be a set of already union-merged sources and $S \notin P$ be the source to be added.

1) $\forall S_j \in P$: S and S_j are disjoint
$\Rightarrow c(P \sqcup S) = c(P) + c(S)$
2) $\forall S_j \in P$: S and S_j are independent
$\Rightarrow c(P \sqcup S) = c(P) + c(S) - c(P) \cdot c(S)$
3) $\exists S_j \in P, S \subseteq S_j$
$\Rightarrow c(P \sqcup S) = c(P)$

□

We briefly discuss the statements of the individual cases of Corollary 1.

1) Case 1 (disjointness): Adding a source to a set that is disjoint to all sources already queried, provides the highest coverage gain. To calculate overall coverage, we simply add the individual scores.
2) Case 2 (independence): To determine the overall coverage we add the scores and subtract the probable overlap between the new source and the already queried sources. Due to the independence assumption of this case, we can quantify this overlap as the product of the two scores.
3) Case 3 (subset/equivalence): When the new source is a subset or equal to one already queried, it does not contribute to coverage in any way. However, it might still be worthwhile to query such a source, as it may well contribute to the overall density score (see below).

If none of the cases applies, coverage calculation is more complicated. Suppose some S_i has mixed overlaps with different sources. These sources in turn may also have mixed overlaps among them. Thus, calculation of the overall coverage score is not straight-forward as in the previous cases, but must be performed recursively as stated in Theorem 1. Note that Theorem 1 includes cases 1 and 2 of Corollary 1. To apply the theorem for coverage calculation, one must first identify the sets of disjoint (D), independent (I), and subset sources (SB). For the independent sources and the subset sources we must calculate $c(I)$, $c(SB)$, and $c(I \sqcap SB)$. The first two terms can be determined again using Theorem 1 in a recursive manner. The last term can be solved with the help of Lemma 2 and Lemma 3:

Lemma 2 ($c(S_i \sqcap S_j)$): Let S_i and S_j be two sources to be join-merged. We distinguish the following cases:

1) S_i and S_j are disjoint
$\Rightarrow c(S_i \sqcap S_j) = 0$
2) S_i and S_j are independent
$\Rightarrow c(S_i \sqcap S_j) = c(S_i) \cdot c(S_j)$
3) $S_i \subseteq S_j$
$\Rightarrow c(S_i \sqcap S_j) = c(S_i)$

□

Lemma 3 ($c(P \sqcap S)$): Let $P = \{S_1, \ldots, S_k\}$ be a set of union-merged sources and $S \notin P$ be the source to be join-merged. Let D be the set of sources in P to which S is disjoint. Let I be the set of sources in P to which S is independent. Let SB be the set of sources in P that are subsets of S. If there are no supersets of S in P, i.e., $\nexists S_j \in P, S \subseteq S_j$, then

$$c(P \sqcap S) = c(S) \cdot c(I) + c(SB) - c(I \sqcap SB).$$

If there is a superset of S in P, i.e., $\exists S_j \in P, S \subseteq S_j$ then

$$c(P \sqcap S) = c(S).$$

□

Note that the set D of sources disjoint to S does not appear in this result, as their content is not part of the result of $P \sqcap S$.

Theorem 1 (Multiple source coverage): Let $P = \{S_1, \ldots, S_k\}$ be a set of already union-merged sources and let $S \notin P$ be the source to be added. Then

$$c(P \sqcup S) = c(P) + c(S) - c(P \sqcap S).$$

□

The theorem is best illustrated as a Venn-diagram as in Figure 5. Source S is to be added, the other sets represent sources already in P. Some of them are disjoint to S (D), some of them are independent (I), and some are subsets (SB). Intuitively, the calculation of coverage first adds the coverage scores of P and S and then subtracts parts that are counted twice. Finally, the parts that are subtracted twice must be added again.

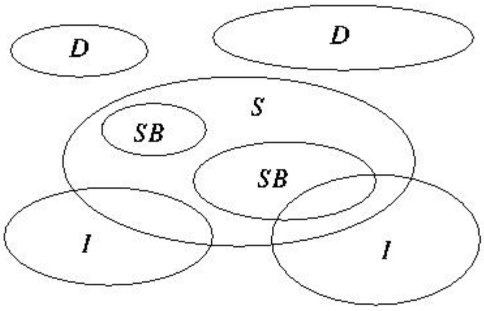

Fig. 5. A Venn-diagram to illustrate coverage calculation

Example. Assume that Merrill Lynch (M) and e*trade (E) are independent sources for stock quotes. Their coverage scores are 0.158 and 0.239, respectively. Thus with Theorem 1, the coverage of the union-merge of the two sources is $0.158 + 0.239 - 0.158 \cdot 0.239 = 0.359$. Assume further that the Yahoo finance (Y) stock information system is (i) independent of e*trade and (ii) a superset of Merrill Lynch; any stock listed by Merrill Lynch is also listed by Yahoo finance. We can then calculate the coverage of the union-merge of all three sources as

$$\begin{aligned}c(M \sqcup E \sqcup Y) &= c(M \sqcup E) + c(Y) - c(Y) \cdot c(E)\\ &\quad - c(M) + c(E \sqcap M)\\ &= 0.359 + 0.25 - 0.25 \cdot 0.239\\ &\quad - 0.158 + c(E \sqcap M)\\ &= 0.391 + 0.158 \cdot 0.239 = 0.429\end{aligned}$$

To verify, we can show that the final score is equal to the coverage of e*trade and Yahoo alone ($c(E \sqcup Y)$), because the Merrill Lynch source is subsumed by Yahoo. □

IV. Density

Density is a measure for the ratio of non-`null`-values provided by sources[6]. Typically, information sources have many missing values (`null`-values) in the attributes they provide, i.e., sources often export attributes they do not completely cover. For instance, book information sites do not provide reviews for all books, an address information service does not have the email address of all people listed, etc. The missing values result in incomplete results, i.e., tables with `null`-values. First, we define density of attributes and sources; then we proceed as in the previous section and show how to determine density of sets of sources.

A. Density of an attribute and density of a source.

Density is attribute specific, i.e., each attribute provided by a source has a density score. In fact, even attributes not provided by a source have a density score for that source. Thus, before defining the density of a source we define the density of an attribute of a source.

Definition 6 (Density): Let D be a domain and $D^+ = D \cup \{\perp\}$. Let X be a multiset (bag) of values $x \in D^+$. The density of X is $|\{x \in D, x \in X\}|/|\{x \in D^+, x \in X\}|$. □

We apply this definition to measure the density of attributes and sources. In accordance to this definition we can define the density of attribute values of an attribute in a source:

Definition 7 (Attribute density): The density of attribute $a \in A$ in source S ($d_S(a)$) is

$$d_S(a) := \frac{|\{t \in S | t[a] \neq \perp\}|}{|S|}$$

where t are tuples of the real world and A is the global set of attributes. □

Definition 8 (Density vector): The density vector $D(S)$ is the vector of the attribute density scores for each attribute of the global schema. $D(S)$ has length $|A|$. □

Thus, an attribute that has a value for every object of the source has a density of 1. An attribute that is simply not provided by a source has density 0. Attributes for which a source can provide some values have a density score in between.

Example. Consider the Yahoo finance table of Table III and assume for this example that it represents the source in its entirety. The density of an attribute is determined by counting the number of null values ($\perp$) in that column and dividing this by the overall number of rows in the table. Thus, $d(\text{symbol}) = 1$, $d(\text{name}) = 0.9$, etc. These scores are summarized in the density vector $D(\text{Yahoo}) = (1,\ 0.9,\ 1,\ 0.9,\ 0.8,\ 0.8,\ 0.4,\ 0,\ 0)$. For typical stock information systems in the real world, attribute density typically is either 0 or 1, depending on which attributes are part of the output of the source. □

Using Definition 6, we can prove that the overall density of a source is the average density of its attributes:

Theorem 2 (Source density): The density of a source S ($d(S)$) is the average density over all attributes:

$$d(S) = \frac{1}{|A|} \sum_{a \in A} d_S(a)$$

□

Proof: Let the set of all data fields in source S be a bag of values $x \in D^+$. Thus, the size of the bag is $|A| \cdot |S|$. Then

$$\begin{aligned}\frac{1}{|A|} \sum_{a \in A} d_S(a) &= \frac{\sum_{a \in A} |\{t \in S | a \neq \perp\}|}{|A| \cdot |S|}\\ &= \frac{|\{x \in D\}|}{|\{x \in D^+\}|} = d(S)\end{aligned}$$

■

B. Density assessment

Like coverage scores, density scores can be assessed in several different ways, depending on the ability and willingness of the information sources to cooperate. In some cases, information sources readily give away the scores. Statements like "We provide reviews for more than 10 percent of all available books" ($d(\text{review}) = 0.1$) or "All search results include a page size" ($d(\text{size}) = 1$) are not uncommon. As in the latter case, density scores are often 0 or 1. They are 0 whenever a source simply does not provide the corresponding attribute of the global relation. The score is 1 whenever the source always provides information for that attribute. For instance, we always require the ID attribute to have a density score of 1. When exact measurement is not possible, sampling techniques can be applied. Certain amounts of information are retrieved, their density is determined and extrapolated to the density of the source. This score can be updated whenever a new result is retrieved from the source. With this continuous update the density score becomes more accurate over time.

Example. Table IV shows the density vectors of the 7 SIS in our example. The scores were assessed by simply examining the search results of an exemplary query, assuming that the values of this result are available for all other queries (and no others). The overall score is the average density score of the attributes. □

C. Density of a set of sources

As for the coverage score we must determine the density of a set of sources to be able to find the best combination. As discussed in Section II, an object in the combined result of two sources has a value in an attribute if either one or both sources provide some value providing that a resolution

[6] The term *density* is derived from the notion of dense vs. sparse matrices.

Yahoo finance:

symbol	name	ltd	ltq	change	change %	volume	td's high	td's low
ACN	Accenture	10:41 AM	18.07	-0.43	-0.10%	⊥	⊥	⊥
BEAS	BEA Systems	10:42 AM	10.98	-0.18	-0.50%	6,292,500	⊥	⊥
CAJ	Canon	10:30 AM	⊥	⊥	⊥	⊥	⊥	⊥
CSCO	Cisco Systems Inc	9:01 AM	14.09	+0.01	+0.05%	47,259,900	⊥	⊥
DELL	Dell Computer Corp	12:00 PM	28.36	-0.25	-1.50%	⊥	⊥	⊥
HPQ	HP	11:11 AM	19.16	+0.01	+0.50%	7,821,800	⊥	⊥
IBM	⊥	12:45 PM	112.20	+1.02	+0.50%	⊥	⊥	⊥
MSFT	Microsoft Corp	13:49 PM	57.89	-0.63	-1.20%	⊥	⊥	⊥
ORCL	Oracle Corp	10:31 AM	11.68	⊥	⊥	29,201,700	⊥	⊥
TOSBF	Toshiba	9:15 AM	4.35	+0.45	+3.04%	⊥	⊥	⊥

TABLE III

THE YAHOO FINANCE STOCK INFORMATION SOURCE

SIS	**overall**	symbol	name	ltd	ltq	change	ch.%	vol.	td's high	td's low
Yahoo fin.	6/9	1	0	1	1	1	1	1	0	0
Yahoo Fin.	7/9	1	1	1	1	1	1	1	0	0
CNN	7/9	1	1	0	1	1	0	1	1	1
NYSE	9/9	1	1	1	1	1	1	1	1	1
e*trade	7/9	1	1	0	1	1	0	1	1	1
AltaVista	8/9	1	1	0	1	1	1	1	1	1
Merrill Lynch	7/9	1	1	0	1	1	0	1	1	1

TABLE IV

DENSITY SCORES FOR STOCK INFORMATION SYSTEMS.

function does not nullify not-null results. As before, we first distinguish several special cases before proving the general result (again S is an information source and P is a set of already merged sources):

- $d_{S_i \sqcap S_j}(a)$ for different overlap cases (Lemma 4)
- $d_{P \sqcap S}(a)$ for different overlap cases (Corollary 2)
- $d_{S_i \sqcup S_j}(a)$ for different overlap cases (Lemma 5)
- $d_{P \sqcup S}(a)$ for different overlap cases (Corollary 3)
- $d_{P \sqcup S}(a)$ for the general case (Theorem 3)

Please note that the individual overlap cases refer to the objects and not to the attribute values of objects. Our definition of overlap concerns only object IDs. For an object represented in more than one source, we do not require the same attribute values or even the same attributes in each source.

Lemma 4 ($d_{S_i \sqcap S_j}(a)$): Let S_i and S_j be the two sources to be join-merged and let a be an attribute of the global schema. We distinguish the following cases:

1) S_i and S_j are disjoint: Because the intersection of two disjoint sources is the empty set, we do not define density of an attribute.
2) S_i and S_j are independent: $d_{S_i \sqcap S_j}(a) = d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)$
3) $S_i \supseteq S_j$ (same as previous case): $d_{S_i \sqcap S_j}(a) = d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)$

□

The proofs of Lemma 4 and the following Lemma 5 are straightforward by applying definitions and performing algebraic transformations. For details, see [11].

Corollary 2 ($d_{P \sqcap S}(a)$): Let $P = \{S_1, \ldots, S_k\}$ be a set of already union-merged sources and $S \notin P$ be the source to be join-merged. Then

$$d_{P \sqcap S}(a) = d_P(a) + d_S(a) - d_P(a) \cdot d_S(a).$$

□

With the help of Lemma 4 and its Corollary 2 we can prove the following lemma and Theorem 3 (again, the proofs can be found in [11]).

Lemma 5 ($d_{S_i \sqcup S_j}(a)$): Let S_i and S_j be the two sources to be union-merged ($S_i \sqcup S_j$). Let the `null`-values of attribute a be distributed independently. We distinguish the following three cases:

1) S_i and S_j are disjoint

$$\Rightarrow d_{S_i \sqcup S_j}(a) = \frac{d_{S_i}(a) \cdot c(S_i) + d_{S_j}(a) \cdot c(S_j)}{c(S_i) + c(S_j)}$$

2) S_i and S_j are independent

$$\begin{aligned} \Rightarrow d_{S_i \sqcup S_j}(a) = \big[& d_{S_i}(a) \cdot c(S_i) + d_{S_j}(a) \cdot c(S_j) \\ & - [d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)] \\ & \cdot c(S_i) \cdot c(S_j) \big] \\ & \cdot \frac{1}{c(S_i) + c(S_j) - c(S_i) \cdot c(S_j)} \end{aligned}$$

3) $S_i \supseteq S_j$

$$\begin{aligned} \Rightarrow d_{S_i \sqcup S_j}(a) = \big[& d_{S_i}(a) \cdot c(S_i) + d_{S_j}(a) \cdot c(S_j) \\ & - [d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)] \\ & \cdot c(S_j) \big] \cdot \frac{1}{c(S_i)} \end{aligned}$$

□

Corollary 3 ($d_{P \sqcup S}(a)$): Let $P = \{S_1, \ldots, S_k\}$ be a set of already union-merged sources and $S \notin P$ be the source to be added. Let the null-values of attribute a be distributed independently. We distinguish the following three cases:

1) $\forall S_j \in P$: S and S_j are disjoint

$$\Rightarrow d_{P \sqcup S}(a) = \frac{d_P(a) \cdot c(P) + d_S(a) \cdot c(S)}{[c(P) + c(S)]}$$

2) $\forall S_j \in P$: S and S_j are independent

$$\begin{aligned}\Rightarrow d_{P \sqcup S}(a) = \big[& d_P(a) \cdot c(P) + d_S(a) \cdot c(S) \\ & - [d_P(a) + d_S(a) - d_P(a) \cdot d_S(a)] \\ & \cdot c(P) \cdot c(S)\big] \\ & \cdot \frac{1}{c(P) + c(S) - c(P) \cdot c(S)}\end{aligned}$$

3) $\exists S_j \in P, S \subseteq S_j$

$$\begin{aligned}\Rightarrow d_{P \sqcup S}(a) = \big[& d_P(a) \cdot c(P) + d_S(a) \cdot c(S) \\ & - [d_P(a) + d_S(a) - d_P(a) \cdot d_S(a)] \\ & \cdot c(S)\big] \cdot \frac{1}{c(P)}\end{aligned}$$

□

This corollary leads us to the general theorem for density.

Theorem 3 (Multiple source attribute density): Let $P = \{S_1, \ldots, S_k\}$ be a set of already union-merged sources and let $S \notin P$ be the source to be added. Let D be the set of sources in P to which S is disjoint. Let I be the set of sources in P to which S is independent. Let SB be the set of sources in P that are subsets of S. Then

$$\begin{aligned}d_{P \sqcup S}(a) = & [d_P(a)c(P) + d_S(a)c(S) - d_{SB}(a)c(SB) \\ & - [d_S(a) + d_I(a) - d_S(a) \cdot d_I(a)]c(S)c(I) \\ & + [d_I(a) + d_{SB}(a) - d_I(a) \cdot d_{SB}(a)] \\ & \cdot c(I \sqcap SB)] \cdot \frac{1}{c(P \sqcup S)}\end{aligned}$$

□

Example. Assume again that Merrill Lynch (M) and e*trade (E) are independent sources. Let the density scores for the *name* attribute (n) be 0.9 and 0.1, respectively. The coverage scores are those used in the previous example (0.158 and 0.239). Thus, the density of their merged result is

$$\begin{aligned}d_{M \sqcup E}(n) = \ & 0.9 \cdot 0.158 + 0.1 \cdot 0.239 \\ & - (0.9 + 0.1 - 0.9 \cdot 0.1) \cdot 0.158 \cdot 0.239 \\ & \cdot \frac{1}{0.158 + 0.239 - 0.158 \cdot 0.239} = \ 0.395\end{aligned}$$

We add the Yahoo finance (Y) SIS and again assume it is independent of e*trade and a superset of Merrill Lynch. We assume that Yahoo has a density of 1 for the name attribute and a coverage of 0.25. The new density of the three for the name attribute is

$$\begin{aligned}d_{M \sqcup E \sqcup Y}(n) = \ & [d_{M \sqcup E}(n)c(M \sqcup E) + d_Y(n)c(Y) \\ & - d_M(n)c(M) \\ & - [d_Y(n) + d_E(n) - d_Y(n) \cdot d_E(n)] \\ & \cdot c(Y)c(E) \\ & + [d_E(n) + d_M(n) - d_E(n) \cdot d_M(n)] \\ & \cdot c(E \sqcap M)] \cdot \frac{1}{c(M \sqcup E \sqcup Y)} \\ = \ & [0.395 \cdot 0.359 + 1 \cdot 0.25 - 0.9 \cdot 0.158 \\ & - [1 + 0.1 - 1 \cdot 0.1] \cdot 0.25 \cdot 0.239 \\ & + [0.1 + 0.9 - 0.1 \cdot 0.9] \cdot 0.038] \cdot \frac{1}{0.429} \\ = \ & 0.523\end{aligned}$$

I.e., when we merge the three sources we can expect to find a name value in over 52 percent of the tuples. □

V. Completeness

The *completeness* of an information source is the ratio of its information amount and the total information of the real world. We understand the amount of information a source can deliver as the number of fields of the global relation it can fill with non-null-values. The more complete a source is, the more information it can potentially contribute to the overall response to a user query.

Definition 9 (Completeness): A source S has completeness

$$C(S) := \frac{\text{number of data-values} \neq \perp \text{ in } S}{|W| \cdot |A|}$$

□

To calculate completeness of an information source without actually counting the number of filled fields, we use coverage and density scores of the source. They are combined in a very natural way:

Theorem 4 (Completeness): Let S be an information source and let $c(S)$ and $d(S)$ be its coverage and density scores, respectively. Then

$$C(S) = c(S) \cdot d(S)$$

□

Corollary 4: Let P be a set of information sources. Then

$$C(P) = c(P) \cdot d(P).$$

□

Example. Suppose Table V represents the entire Yahoo finance information source, i.e., it provides only two tuples with varying density. Coverage of the source is thus $c(\text{Yahoo}) = 1/20,000$. The density vector of the source is $D(\text{Yahoo}) = (1, 0, 1, 1, 1, 1, 1, 0, 0)$ and the density is $d(\text{Yahoo}) = 2/3$. Thus, with Theorem 4 completeness of Yahoo finance (in this miniature example) is $1/20,000 \cdot 2/3 = 1/30,000$. This number corresponds to the definition of completeness: The number of non-null values in the source is 12 and $|W| \cdot |A| = 40,000 \cdot 9 = 360,000$ and $12/360,000 = 1/30,000$. □

Yahoo Finance

symbol	name	ltd	l. tr. quote	change	change %	volume	td's high	td's low
IBM	$\perp$	10:45 AM	112 $1/8$	$+9/16$	+0.50%	1,458,600	$\perp$	$\perp$
IBM SICO.	$\perp$	9:47 AM	111	$+8/16$	+1.2%	677	$\perp$	$\perp$

TABLE V
AN INFORMATION SOURCE

Theorem 4 and Corollary 4 suggest that completeness calculation can be interpreted as the geometric calculation of an area: Coverage represents the height of the area (or table), density represents the width of the area (or table). In the following section we suggest several applications for the completeness measure and provide an outlook to future research.

VI. CONCLUSIONS AND OUTLOOK

Our coverage, density, and completeness models are a powerful tool with several applications in cooperative information systems. Among them is the task of *source selection* and *plan selection* as described in [12]: When trying to decide which source or set of sources to query, our model offers an excellent guideline for chosing the most promising set of sources based on the expected information quality. For instance, the coverage criterion is of special importance when comparing search engines. One of the main features of search engines is the amount of Web pages they have previously indexed. The larger a search engine, the more probable it is to find the desired result. Coverage calculation corresponds to join-result size estimation in traditional database systems. Other application domains demand special attributes to perform joins. The density measure is well suited to select sources on this basis. The completeness measure combines the two; it provides hints on the byte-size of the result – an important measure for applications with widely distributed data and/or low bandwidth connections between the sources. Taking source selection one step further, completeness measures are useful for selecting the best query execution plan across several sources: Sections III-C and IV-C expand the notion of coverage and density of sources to that of sets of sources or plans. Thus, with the value model we present, a meta information service can generate and compare different strategies to execute a user query against a cooperative information system. The measures of this paper seem to imply that large sources are good sources. High coverage and density are better than low scores. On the other hand, much has been lamented on the *information overflow* caused by the enormous size of the World Wide Web. Much research has addressed the problem of reducing query responses to a reasonable number of objects, if possible to the most useful or relevant ones to the user. This need for reduction is especially true for search engines, where no user is willing to browse the typical number of $> 10,000$ results. However, any filtering profits from a large amount of information to begin with. The model presented in this paper is able to objectively value information sources by the amount of information they provide.

We consider our model as an important step towards the systematic consideration of information quality in data integration. We plan to use our approach in projects in the area of bioinformatics. In bioinformatics, sources typically contain a set of core attributes of high accuracy, describing their primary data objects, and an extended set of other attributes that are not updated on a regular basis. Hence, choosing the right source dependent on the attributes being seeked is an essential problem. We believe that our density, coverage, and completeness measures provide a solid ground for this task.

VII. ACKNOWLEDGEMENTS

This research was partly supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK 316), and the German Ministry for Education and Research (BMBF grant).

REFERENCES

[1] H. Plattner, Keynote at SAP Konferenz für Business Intelligence und Enterprise Portals, Leipzig, Germany, Jan 2002.

[2] Y. Chen, Q. Zhu, and N. Wang, "Query processing with quality control in the World Wide Web," *World Wide Web*, vol. 1(4), pp. 241–255, 1998.

[3] A. Motro and I. Rakov, "Estimating the quality of databases," in *Proceedings of the International Conference on Flexible Query Answering Systems (FQAS)*. Roskilde, Denmark: Springer Verlag, May 1998, pp. 298–307.

[4] D. Florescu, D. Koller, and A. Levy, "Using probabilistic information in data integration," in *Proceedings of the International Conference on Very Large Databases (VLDB)*, Athens, Greece, 1997, pp. 216–225.

[5] D. Maier, J. D. Ullman, and M. Y. Vardi, "On the foundations of the universal relation model," *ACM Transactions on Database Systems (TODS)*, vol. 9(2), pp. 283–308, 1984.

[6] R. Yerneni, Y. Papakonstantinou, S. Abiteboul, and H. Garcia-Molina, "Fusion queries over internet databases," in *Proceedings of the International Conference on Extending Database Technology (EDBT)*, Valencia, Spain, Mar. 1998.

[7] M. Neiling, S. Jurk, H.-J. Lenz, and F. Naumann, "Object identification quality," in *Proceedings of the International Workshop on Data Quality in Cooperative Information Systsems (DQCIS)*, Siena, Italy, 2003.

[8] C. Yu and W. Meng, *Principles of database query processing for advanced applications*. San Francisco, CA, USA: Morgan Kaufmann, 1998.

[9] C. Date, *Relational Database (selected writings)*. Reading, MA, USA: Addison-Wesley, 1986.

[10] M. LaCroix and A. Pirotte, "Generalized joins," *SIGMOD Record*, vol. 8(3), pp. 14–15, September 1976.

[11] F. Naumann and J. C. Freytag, "Completeness of information sources," Humboldt-Universität zu Berlin, Institut für Informatik, Tech. Rep. 135, 2000.

[12] F. Naumann, *Quality-driven Query Answering for Integrated Information Systems*, ser. Lecture Notes on Computer Science (LNCS). Heidelberg: Springer Verlag, 2002, vol. 2261.