Completeness of Information Sources

Felix Naumann Johann Christoph Freytag
Humboldt-Universität zu Berlin
{naumann|freytag}@dbis.informatik.hu-berlin.de

Abstract

For a large number of information domains, there are numerous World Wide Web information sources. Mediators allow integrated access to these sources by providing a common schema against which the user can pose queries. The sources often vary both in their extension and their intension: Due to their autonomy, sources overlap in the objects they cover and differ in the attributes of the objects they provide.

We support mediators in their source selection and query planning process by a value model that incorporates both extensional value (coverage) and intensional value (density). When results from sources are merged, the scores for coverage and density of the merged result must be estimated. For both criteria we provide *merge functions* that calculate the score of merged results. Also, we combine the two criteria to an overall *completeness* criterion. This completeness measure is a valuable tool to assess source size and to predict result sizes of queries in WWW settings.

1 An Information Source Value Model

The paradigm of information querying has dramatically changed in recent years. While users assume a centralized database management system to always have all the information, the merit of such a system is measured by its speed or response time. The assumption of completeness no longer applies to WWW information sources. The speed or response time of an Internet source is secondary compared to its ability to provide information as a result of a query. For example, a query for stock information to any given WWW stock information system will no longer return all the information there is for a certain stock. Some sources will provide no company profile, others will provide day-old information, again others simply may not provide any information for that stock at all. Or inversely, some sources provide additional information that others do not.

To gain the full advantage of the Internet, a user should query all available sources and integrate the results. This is a tedious and often expensive task. Developers can and already have designed systems that automatically perform this task but this still does not reduce the cost of querying all sources. Especially in the case of stock information systems, queries often result in paying a certain amount of money. 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 stock information systems this is 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.).

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 [CZW98]. Motro and Rakov define a "completeness" criterion [MR98]. 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 size of information sources. To the best of our knowledge the density criterion as we define it has never before been addressed in literature.

Structure of this paper. What follows is a thorough definition and analysis of the two criteria coverage and density. For each criterion we will show how to merge scores of different sources with varying overlap situations. Finally, we combine the two criteria into the general completeness criterion. The stock information system example will guide the reader through our approach.

2 Web Querying

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

2.1 The information model

- Global schema: We assume a global schema that consists of only one relation. This relation contains the global 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 which projects out any attribute not delivered by the source. However, each source must deliver the ID attribute. We assume heterogeneity to be resolved inside the wrappers (see below).

  Again, having only one global relation seems overly restrictive, but is in many cases a convenient and sufficient model. However, it is not suitable if complex object relationships need to be modeled.

- 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 [YPAGM98].

  While this may seem like a strong assumption, 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. A person ID could be the combined name and address fields.

- Overlap: We assume that source contents overlap to various degrees. In an extreme case one source can be a mirror of another source, i.e., they totally overlap. Other degrees of overlap are
  - containment: The IDs in one source are a subset of the IDs in another source. I.e., the contained source only stores information about objects that the other source also stores information about. However, the actual information, i.e., the attribute values might differ.
  - independence: There is no (known) dependency between the IDs of the sources. Whenever there is no knowledge about containment or disjointness, we assume independence. I.e., there is some coincidental overlap, determined by the size of the sources and the size of the real world they model.
  - 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 with more attribute values.

Example. We will use a meta stock information system (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 following result[1]:

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

Yahoo finance in its basic form 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 provided by the set of the following 7 SIS as a global schema. Whenever an attribute is not provided by a source, the corresponding field will be left empty (`null`-value).

- 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`

[1] We used the Yahoo finance system for this example.

[2] The detailed table provides many more fields.

- AltaVista Money section `money.altavista.com`
- Merrill Lynch `www.ml.com`

Our global relation for SIS results is shown in Table 1. The global IDs of SIS results are the stock *symbol*. The name 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. Some SIS provide much more information such as company profiles, charts etc. For simplicity we ignore these in this report.

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

Table 1: Global relation for search engine query results

The results of SIS queries typically overlap, i.e., two systems may return information for the same stock or symbol. However, the attributes of the results 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. □

2.2 Result merging

An MSIS distributes a user query to multiple information systems. After receiving the individual results, it is the task of the MSIS 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 will contain objects where

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

Result merging of the MSIS in the first case is clear – the object in the result will have 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. 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 *resolution function* must determine what value will appear in the result table.

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

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

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.

Resolution functions can be of various types, depending on the type of attribute, the usage of the value and many other aspects [YM98][3]. One simple resolution function might concatenate the values and annotate them with the source that provided the value. Especially conflicts in textual attributes can be resolved in this way. A resolution function for numerical values might be to determine the average value.

To formalize the merging approach we define new operators, the join-merge-operator denoted $\sqcap$ and the union-merge-operator denoted $\sqcup$. Both operators include resolution functions in case of data conflicts. First, we define the join-merge-operator and show an example in Figure 1.

Definition 2 (Join-Merge Operator $\sqcap$). *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. The attributes $A_i, \dots, A_m$ are common in both relations, i..e., they are each mapped to the same attribute in the global schema. Then*

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

where $f()$ is a resolution function 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	$\perp$	$f(z, y)$	$\perp$

Figure 1: The join-merge-operator

Definition 3 (Union-Merge Operator $\sqcup$). *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*

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

The union-merge operator is an extension of the join-merge operator. The union-merge-operator guarantees that every tuple enters a join. An example is shown in Figure 2.

We call this operator *merge* operator 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.

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

$r:$	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	$\perp$	$f(z,y)$	$\perp$
	4	$\perp$	x	i

Figure 2: The union-merge-operator

LaCroix and Pirotte defined a similar operator, the "generalized natural join operator", denoted $\overset{+}{\bowtie}$ [LP76]. 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[4].

Example. Imagine two SIS delivering some results to a query for "IBM". Some of the results will be common, however with differing attributes and different attribute values, some results will be distinct. In order not to lose any information, the results are merged in the result table. Table 2 shows the two search results and the merged result for the user.

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	112 1/8	+9/16	+0.50%	1,529,500	112 13/16	111
IBM SICO.	$\perp$	9:47 AM	111	+8/16	+1.2%	677	$\perp$	$\perp$

Table 2: Two results (Yahoo finance and CNN) and the merged response

Observe that the first line 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 this is the more recent information and we choose that value.

□

3 Coverage

We define *coverage* of a source to be the number of objects that a source can potentially return – 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

[4] Other interpretations of the merge operator are a variation of the two-way outer-join or the union operator

that the set can potentially return. Since 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 connection 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.

3.1 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 real world (number of real objects):

Definition 4 (The World). *Given a global relation of an application domain, we define the world W as the set of all real world objects that pertain at least in parts to the global relation. The number of real world objects is $|W|$.*

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 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.

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

Stock information system	**Number stocks listed**	**Coverage 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 3: Stock information system coverage

3.2 Coverage and overlap assessment

The coverage measure for 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 will publish coverage scores as a means for advertising their service. However, not always can these figures be trusted. Another possibility is to simply measure coverage, where possible. This may be possible by downloading the source or querying the source. If these assessment methods fail, coverage score can only be estimated.

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 both list. 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).

□

3.3 Coverage of a set of sources.

To respond to a user query in the best possible way, a query can 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. In particular, we show how to calculate the coverage of the following terms, where S is a source 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.

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

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:*

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. $S_i \subseteq S_j$ $\Rightarrow c(S_i \sqcup S_j) = c(S_j)$

Proof.

$$1. \quad c(S_i \sqcup S_j) = \frac{|S_i \sqcup S_j|}{|W|} = \frac{|S_i| + |S_j|}{|W|} = \frac{|S_i|}{|W|} + \frac{|S_j|}{|W|} = c(S_i) + c(S_j)$$

$$2. \quad c(S_i \sqcup S_j) = \frac{|S_i \sqcup S_j|}{|W|} = \frac{|S_i| + |S_j| - |S_i \sqcap S_j|}{|W|} = \frac{|S_i|}{|W|} + \frac{|S_j|}{|W|} - \frac{|S_i \sqcap S_j|}{|W|}$$
$$= c(S_i) + c(S_j) - c(S_i) \cdot c(S_j)$$

$$3. \quad c(S_i \sqcup S_j) = \frac{|S_i \sqcup S_j|}{|W|} = \frac{|S_j|}{|W|} = c(S_j)$$

□

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. $\exists S_j \in P, S \subseteq S_j$ $\Rightarrow c(P \sqcup S) = c(P)$

Discussion. 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. 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, independent, and subset sources. 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 $P = \{S_1, \dots, S_k\}$ 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., $\nexists S_j \in P, S \subseteq S_j$*[6]*, then*

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

Proof. Because $\nexists S_j \in P, S \subseteq S_j$, we can partition the sources of P into D, I, and SB as described above.

$$\begin{aligned} c(P \sqcap S) &= c((D \sqcup I \sqcup SB) \sqcap S) \\ &= c((D \sqcap S) \sqcup (I \sqcap S) \sqcup (SB \sqcap S)) \\ &= c((S \sqcap I) \sqcup (S \sqcap SB)) \\ \text{Corollary 1} &= c(S \sqcap I) + c(S \sqcap SB) - c(S \sqcap I \sqcap SB) \\ \text{Lemma 2} &= c(S) \cdot c(I) + c(SB) - c(I \sqcap SB) \end{aligned}$$

□

Theorem 1 (Multiple source coverage). *Let $P = \{S_1, \dots, S_k\}$ 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 in Figure 3. 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). The assumptions of Theorem 1 exclude the existence of any supersets of S. Querying a source S of which a superset has already been queried makes no sense. Intuitively, the calculation of coverage first adds the

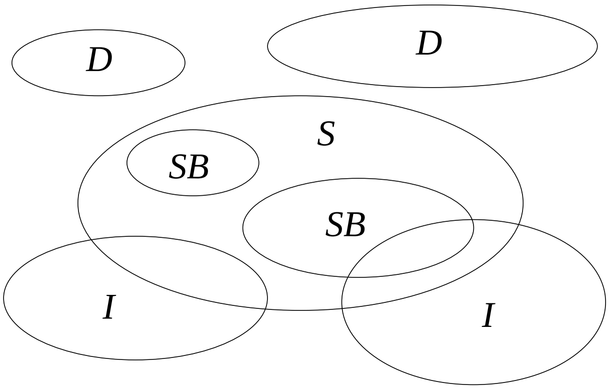

Figure 3: A Venn-diagram to illustrate coverage calculation

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[7].

Example. Assume that Merrill Lynch (M) and e*trade (E) are independent sources. Their coverage scores are 0.158 and 0.239 respectively. Thus, together their coverage is $0.158 + 0.239 - 0.158 \cdot 0.239 = 0.359$. If we add the Yahoo finance (Y) SIS and assume (for the sake of the example) it is independent of e*trade and a superset of Merrill Lynch, i.e., any stock listed by Merrill Lynch is also listed by Yahoo finance. The new coverage of the three is

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

□

4 Density

Density is a measure for the ratio of non-`null`-values provided by sources[8]. Typically, information sources may have many missing values (`null`-values) in the attributes they provide, i.e., sources often provide attributes they do not completely cover. For instance, book information sites do not provide reviews for all books, an address information service will 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 and the proceed as in the previous section and show how to determine density of sets of sources.

4.1 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 will 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.

[6] Is there is a superset of S in P, i.e., $\exists S_j \in P, S \subseteq S_j$ then $c(P \sqcap S) = c(S)$.

[7] With this explanation we choose to spare the proof.

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

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 $|\{x \in D\}|/|\{x \in D^+\}|$.*

We apply this definition to measure the density of attributes (no proof) and sources. In accordance to this definition we can define the density of attribute values of of an attribute in a source:

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

$$d_S(a) := \frac{|\{t \in S | a \neq \perp\}|}{|S|} \text{ or in short hand } d_S(a) := \frac{nn(S)}{|S|}$$

where t are tuples of the real world and A is the global set of attributes. 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 will have a density of 1. An attribute that is simply not provided by a source will have density 0. We define density of a source as 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 fields in source S be a bag of values $x \in D^+$. Thus, the size of the bag is $|A| \cdot |S|$. Then

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

□

Example. For SIS attribute density typically is either 0 or 1, depending on the output format of the source. For instance, the density of the CNN SIS is $d(\text{CNN}) = 7/9$ and its density vector is $D(\text{CNN}) = (1, 1, 0, 1, 1, 0, 1, 1, 1)$, i.e., the CNN SIS returns Symbol, name, last trade quote, change, volume, and today's high and low information for *every* stock and the other two attributes (last trade date and change %) for *no* stock. Other SIS will have more variant density scores. For instance, some stock information service may have company profiles only for *some* companies, online book stores only provide reviews for an assortment of books. □

4.2 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 will readily give away the scores. Statements like "We provide reviews for more than

10% 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 to that attribute. For instance, we assume this is always the case for the ID attribute.

When direct methods are 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 will become more accurate over time.

Example. Table 4 shows the density vectors of the 7 SIS in our example. The scores were assessed by simply examining the search results. The overall score is the average density score of the attributes.

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 4: Density scores for search engines

□

4.3 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 2, an object in the combined result of two sources will have a value in an attribute if either one or both sources provide some value. 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. We assume that any overlap, be it due to independence or containment, only concerns the object. 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. We distinguish the following cases:*

1. *S_i and S_j are disjoint: Since 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):*

$$d_{S_i \sqcap S_j}(a) = d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)$$

Proof.

2. S_i and S_j are independent (recall that $nn(S_i \sqcap S_j) := |\{t \in S_i \sqcap S_j | a \neq \bot\}|$):

$$\begin{aligned} d_{S_i \sqcap S_j}(a) &= \frac{nn(S_i \sqcap S_j)}{|S_i \sqcap S_j|} \\ &= \frac{|W| \cdot [c(S_i)c(S_j)d_{S_i}(a) + c(S_i)c(S_j)d_{S_j}(a) - c(S_i)c(S_j)d_{S_i}(a)d_{S_j}(a)]}{|W| \cdot c(S_i) \cdot c(S_j)} \\ &= d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a) \end{aligned}$$

3. $S_i \supseteq S_j$ (slightly different proof but same result as previous case):

$$\begin{aligned} d_{S_i \sqcap S_j}(a) &= \frac{nn(S_i \sqcap S_j)}{|S_i \sqcap S_j|} \\ &= \frac{|W| \cdot [c(S_j)d_{S_i}(a) + c(S_j)d_{S_j}(a) - c(S_j)d_{S_i}(a)d_{S_j}(a)]}{|W| \cdot c(S_j)} \\ &= d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a) \end{aligned}$$

□

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.

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 ($c(S)$ is short for $c(S)$):*

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. $S_i \supseteq S_j$

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

Proof.

1. S_i and S_j are disjoint

$$\begin{aligned}d_{S_i \sqcup S_j}(a) &= \frac{nn(S_i \sqcup S_j)}{|S_i \sqcup S_j|} \\ &= \frac{nn(S_i) + nn(S_j)}{|S_i| + |S_j|} \\ &= \frac{nn(S_i)/|W| + nn(S_j)/|W|}{|S_i|/|W| + |S_j|/|W|} \\ &= \frac{nn(S_i) \cdot |S_i|/|W| \cdot |S_i| + nn(S_j) \cdot |S_j|/|W| \cdot |S_j|}{c(S_i) + c(S_j)} \\ &= \frac{d_{S_i}(a) \cdot c(S_i) + d_{S_j}(a) \cdot c(S_j)}{c(S_i) + c(S_j)}\end{aligned}$$

2. S_i and S_j are independent

$$\begin{aligned}d_{S_i \sqcup S_j}(a) =& \frac{nn(S_i \sqcup S_j)}{|S_i \sqcup S_j|} \\ =& \frac{nn(S_i) + |\{t \in S_j | a \neq \bot\}| - nn(S_i \sqcap S_j)}{c(S_i \sqcup S_j) \cdot |W|} \\ =& \frac{nn(S_i)/|W| + nn(S_j)/|W| - nn(S_i \sqcap S_j)/|W|}{c(S_i) + c(S_j) - c(S_i) \cdot c(S_j)} \\ \text{(Lemma 4)} =& \big[nn(S_i)/|W| + nn(S_j)/|W| - |W| \cdot [d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)] \\ & \cdot c(S_i) \cdot c(S_j)/|W|\big] \cdot \frac{1}{c(S_i) + c(S_j) - c(S_i) \cdot c(S_j)} \\ =& \big[d_{S_i}(a) \cdot c(S_i) + d_{S_j}(a) \cdot c(S_j) - [d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)] \\ & \cdot c(S_i) \cdot c(S_j)\big] \cdot \frac{1}{c(S_i) + c(S_j) - c(S_i) \cdot c(S_j)}\end{aligned}$$

3. For $S_i \supseteq S_j$ the proof is similar to the independent case, only the denominator changes.

$$\begin{aligned}
d_{S_i \sqcup S_j}(a) =& \frac{nn(S_i \sqcup S_j)}{|S_i \sqcup S_j|} \\
=& \frac{nn(S_i) + nn(S_j) - nn(S_i \sqcap S_j)}{c(S_i) \cdot |W|} \\
\text{(Lemma 4)} =& \big[nn(S_i)/|W| + nn(S_j)/|W| \\
& - |W| \cdot [d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)] \cdot c(S_i) \cdot c(S_j)/|W|\big] \cdot \frac{1}{c(S_i)} \\
=& \big[d_{S_i}(a) \cdot c(S_i) + d_{S_j}(a) \cdot c(S_j) \\
& - [d_{S_i}(a) + d_{S_j}(a) - d_{S_i}(a) \cdot d_{S_j}(a)] \cdot c(S_i) \cdot c(S_j)\big] \cdot \frac{1}{c(S_i)}
\end{aligned}$$

□

Corollary 3 ($d_{P \sqcup S}(a)$). *Let $P = \{S_1, \dots, S_k\}$ 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 ($c(S)$ is short for $c(S)$):*

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. $\exists S_j \in P, S \subseteq S_j$

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

This leads us to the general theorem for density.

Theorem 3 (Multiple source attribute density). *Let $P = \{S_1, \dots, S_k\}$ 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*

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

Proof.

$$\begin{aligned}
d_{P\sqcup S}(a) =& \frac{nn(P \sqcup S)}{|P \sqcup S|} \\
=& \frac{nn(P) + nn(S) - nn(S \sqcap I) - nn(SB) + nn(I \sqcap SB)}{|W| \cdot c(P \sqcup S)} \\
\text{(Lemma 4)} =& \big[d_P(a) \cdot |P| + d_S(a) \cdot |S| - [d_S(a) + d_I(a) - d_S(a) \cdot d_I(a)] \cdot |S \sqcap I| \\
& - d_{SB}(a) \cdot |SB| + [d_I(a) + d_{SB}(a) - d_I(a) \cdot d_{SB}(a)] \cdot |I \sqcap SB|\big] \cdot \frac{1}{|W| \cdot c(P \sqcup S)} \\
=& \big[|W| \cdot [d_P(a) \cdot c(P) + d_S(a) \cdot c(S) \\
& - [d_S(a) + d_I(a) - d_S(a) \cdot d_I(a)]c(S)c(I) - d_{SB}(a) \cdot c(SB) \\
& + [d_I(a) + d_{SB}(a) - d_I(a) \cdot d_{SB}(a)] \cdot c(I \sqcap SB)\big] \cdot \frac{1}{|W| \cdot c(P \sqcup S)} \\
=& \big[d_P(a) \cdot c(P) + d_S(a) \cdot c(S) \\
& - [d_S(a) + d_I(a) - d_S(a) \cdot d_I(a)]c(S) \cdot c(I) - d_{SB}(a) \cdot c(SB) \\
& + [d_I(a) + d_{SB}(a) - d_I(a) \cdot d_{SB}(a)] \cdot c(I \sqcap SB)\big] \cdot \frac{1}{c(P \sqcup S)}
\end{aligned}$$

□

Example. Assume again that Merrill Lynch (M) and e*trade (E) are independent sources. For the sake of the example assume that the density scores for the name attribute (n) are 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 assume it is independent of e*trade and a superset of Merrill Lynch, i.e., any stock listed by Merrill Lynch is also listed by Yahoo finance. 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 50% of the tuples. □

5 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 8 (Completeness). *A source S has completeness*

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

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*

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

Proof. The proof simply follows from the definitions of coverage and density.

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

□

Corollary 4. *Let P be a set of information sources. Then $C(P) = c(P) \cdot d(P)$.*

Example. Suppose Table 5 represents a whole information source, i.e., that particular information source 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 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	⊥	10:45 AM	112 1/8	+9/16	+0.50%	1,458,600	⊥	⊥
IBM SICO.	⊥	9:47 AM	111	+8/16	+1.2%	677	⊥	⊥

Table 5: 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.

6 Conclusions and Outlook

First, we sketch several application areas for our completeness measure. Then we briefly discuss our future research plans.

Source selection. The coverage, density, and completeness models are a powerful tool with several uses for WWW information system integration. First, the measures can be used for source selection. When trying to decide which source or set of sources to query the models can be a good guideline. 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. For instance, it gives hint on the byte-size of the result – an important measure for applications with widely distributed data and/or low bandwidth connections between the sources.

Plan selection. Sections 3.3 and 4.3 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.

Information overflow. 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 is especially true for search engines, where no user is willing to browse the typical number of $> 10,000$ results. However, any filtering result to present relevant information will profit from a large amount of information to begin with. The model presented in this paper is able to objectively value WWW information sources by the amount of information they provide.

Future research. We plan to apply the coverage, density, and completeness measures to planning algorithms. We will use the measures to formulate optimization problems where the search space is a set of sources or plans. The optimal subset of sources or the optimal plan are defined as the search space element with highest completeness under certain cost constraints. Using this model we plan to develop algorithms that find optimal or near optimal solutions. The measures and algorithms are parts of a planned prototype MSIS.

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

References

[CZW98] Ying Chen, Qiang Zhu, and Nengbin Wang. Query processing with quality control in the World Wide Web. *World Wide Web*, 1 (4):241–255, 1998.

[LP76] M. LaCroix and A. Pirotte. Generalized joins. *ACM SIGMOD Record*, 8:3:14–15, September 1976.

[MR98] Amihai Motro and Igor Rakov. Estimating the quality of databases. In *Proc. of the 3rd Int. Conf. on Flexible Query Answering Systems*, Roskilde, Denmark, May 1998. Springer Verlag.

[YM98] C. Yu and W. Meng. *Principles of database query processing for advanced applications.* Morgan Kaufmann, San Francisco, CA, USA, 1998.

[YPAGM98] Ramana Yerneni, Yannis Papakonstantinou, Serge Abiteboul, and Hector Garcia-Molina. Fusion queries over internet databases. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, Valencia, Spain, March 1998.