On the Effectiveness of Automatic Schema Matching Over Heterogeneous Digital Libraries

M. Elena Renda and Umberto Straccia

ISTI-CNR
Via G. Moruzzi 1, 56124 Pisa, Italy
{elena.renda, umberto.straccia}@isti.cnr.it

Abstract. Schema matching is the problem of finding mappings between different structured schemas. When retrieving information from digital libraries with heterogeneous schemas, the query over a target schema has to be transformed into a query over the schema of the digital library (the source schema) it has been submitted to. Schema mappings define the rules for this query transformation.
In this paper we address the issue of automatically learning these mappings; furthermore, we evaluate their effectiveness in accessing distributed, heterogeneous digital libraries.

1 Introduction

Federated digital libraries integrate a large number of legacy libraries and give users the impression of one coherent, homogeneous library. These libraries use different metadata schemas (called source schemas). As users cannot deal efficiently with this semantic heterogeneity, they only see one system-wide or personalized target (or global, or mediated) metadata schema, which is defined independently of the libraries. In such a context, three major tasks have to be executed to answer to a query q issued by users over the target schema T. Let us assume for the sake of concreteness that queries are of the form

$$q = \{A_1 = v_1, \ldots, A_q = v_q\} \ ,$$

where each A_i is an attribute of the target (global) schema and v_i is a type correct value for attribute A_i. A query example over the metadata schema $\texttt{T}(\texttt{author}, \texttt{abstract})$ may be:

$$q = \{\texttt{abstract} = \text{“}logic\text{”}, \texttt{abstract} = \text{“}computer\ science\text{”}, \texttt{author} = \text{“}Moshe\ Vardi\text{”}\}$$

whose intended meaning is to “retrieve all documents written by Moshe Vardi, which talk about logic in computer science”.

Then, the following tasks have to be executed:

1. *select* a subset of *relevant* (or, most promising) digital libraries among all the digital libraries that can be accessed by a system, as it is not cost-effective to submit a query to all possible resources. This task is called *automated resource selection* in the literature (see, e.g. [3, 13]);

2. for every selected digital library (called *resource*), *reformulate* the information need q into a query $\bar{q}$ over the query language provided by the resource. Query reformulation is based on *schema mappings*, which are usually automatically learnt in the context of the *automated schema matching* task (see, e.g. [26]). These mappings are usually of the form $A_T \rightarrow A_S$, stating that attribute A_T of the target schema can be mapped into the attribute A_S of the source schema. For instance, suppose the source schema is $\texttt{S}(\texttt{creator}, \texttt{description})$, then under the set of mappings $\{\texttt{author} \rightarrow \texttt{creator}, \texttt{abstract} \rightarrow \texttt{description}\}$ the above query will be rewritten as

$$Q = \{\texttt{description} = \text{“}logic\text{”}, \texttt{description} = \text{“}computer\ science\text{”}, \texttt{creator} = \text{“}Moshe\ Vardi\text{”}\}\ ;$$

3. submit the transformed queries to the selected resources and merge the rank lists together. This latter task is called *rank fusion* in the literature (see, e.g. [27]).

In summary, you have to know *where to search*, *how to query different digital libraries*, and *how to combine the retrieved information from diverse resources.* In this paper we deal with the schema matching task, i.e. how to learn the mappings automatically, without human intervention, for querying different digital libraries we are accessing to.

The framework we propose here relies on simple and effective method to automatically learn schema mappings in such a scenario. While automatic schema matching has already been addressed in the literature, the novelty is that we propose a technique based on a resource selection method [5]. We recall that resource selection is the task of identifying relevant libraries for a given query. Now take a resource S and its metadata schema with attributes $S_1, ..., S_n$ (the source schema). The resource selection task can be reformulated in the schema matching problem as follows: given an attribute-value pair $A_i = v_i$, with A_i being an attribute of the target schema, select among all the attributes S_j those which are most relevant to the attribute A_i given its value v_i, and map A_i to the most relevant attribute.

The structure of the paper is the following: Section 2 describes how this work is related to other approaches; Section 3 introduces our formal framework for schema mapping, based on resource selection; in Section 4 we report the evaluation of the proposed approach on two different data sets; and Section 5 concludes.

2 Related Work

With the proliferation of Digital Libraries over the Web, the development of automated tools for schema matching is of particular importance to automatize distributed retrieval.

The matching problem has been addressed by many researchers in two related areas: the matching problem for “database” schemas, and the matching problem

for "ontologies" [1]. These two areas are closely related as, e.g., schemas can be seen as ontologies with restricted relationship types. The techniques applied in schema matching can be applied to ontology matching as well; additionally, we have to take care of the hierarchies.

Related to ontology matching are, for instance, the works [16, 18, 24] (see [10] for an extensive comparison). While most of them use a variety of heuristics to match ontology elements, very few do use machine learning and exploit information in the data instances [10, 16, 18].

Related to schema matching are, for instance, the works [1, 2, 6–9, 11, 12, 14, 15, 17, 19–23, 25, 29] (see [6, 26] for an extensive comparison). Most recent approaches, either implicitly or explicitly, perform schema mapping based on attribute name comparison and/or comparing properties of the underlying data instances using machine learning techniques. In particular, applying machine learning techniques requires instances from both the target schema (global query language) and the source schema. In these cases both the target schema and the source schema are relational database tables. The attribute matching process is based on some comparison between the values in the source table and the target table.

However, these methods do not directly apply to our specific case as we have just some queries over the target schema and not a relational table.

3 Formal Framework for Schema Mapping

The framework we propose here relies on a simple and effective method to automatically learn schema mappings. It is based on a reformulation of the CORI resource selection framework [5]. In the following, after some preliminary definitions, we first describe how CORI works for automated resource selection, and then we fit this method into our context.

Preliminaries. Our framework is based on the following assumptions. We assume that a user query q over the target metadata schema T is based on single-valued attributes $A_1, \ldots, A_q$, i.e. a query is a set of attribute-value pairs of the form $q = \{A_1 = v_1, \ldots, A_q = v_q\}$, where each A_i is an attribute of the schema and v_i is a type correct value for attribute A_i[2]. We further assume that there are n distributed information resources $\mathscr{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$ accessible to the user. Each resource $\mathcal{R}_i$ has schema S_i based on attributes $A_{i_1}, \ldots, A_{i_q}$, and supports two types of queries: *simple queries* and *complex queries*. A simple query is just a set of values $q^s = \{v_1, ..., v_m\}$, while a complex query for resource $\mathcal{R}_i$ is a set of attribute-value pairs $q_i^c = \{A_{i_1} = v_{i_1}, \ldots, A_{i_q} = v_{i_q}\}$ over the schema attributes of $\mathcal{R}_i$.

[1] Informally, an ontology consists of a hierarchical description in some suitable logical language of important concepts in a particular domain, along with the description of the properties of the instances of each concept.

[2] Note that a query attribute may appear in the query multiple times.

Automated resource selection using CORI. Automated resource selection is based on the assumption of having a significant set of records available from each information resource (see, e.g. [3, 13]). Usually, these records are obtained by issuing random queries to the resource. This allows to compute an *approximation of the content* of each information resource. This task is called *information resource sampling* and methods for computing it exists (see, e.g. [4]). Information resource sampling consists of the computation of a representation of what an information resource is about, i.e. the so-called *information resource topics* or *language model* of the information resource. As a result, a *sample* set of records for each information resource is gathered. This set is our *resource description* or *approximation* of the information resource.

This data is then used in the next step to compute the *resource score* for each information resource, i.e. a measure of the relevance of a given resource to the query. The resource score establishes the relatedness of an information resource with respect to the query. The resource score is computed using an adapted version of the CORI resource selection method, which we describe next.

Consider the original query $q = \{A_1 = v_1, \ldots, A_q = v_q\}$. At first, we unfold the complex query q into a simple query $q' = \{v_1, ..., v_q\}$, where the attributes A_i have been removed. Now, for each resource $\mathcal{R}_i \in \mathscr{R}$, we associate the *resource score*, or simply the *goodness*, $G(q, \mathcal{R}_i)$, which indicates the relevance of resource $\mathcal{R}_i$ to the query q. Informally, a resource is more relevant if its approximation, computed by query-based sampling, contains many terms related to the original query. However, if a query term occurs in many resources, this term is not a good one to discriminate between relevant and not relevant resources. The weighting schema is as follows:

$$G(q, \mathcal{R}_i) = \frac{\sum_{v_k \in q'} p(v_k|\mathcal{R}_i)}{|q'|} \ , \tag{1}$$

where $|q'|$ is the number of values in q'. The *belief* $p(v_k|\mathcal{R}_i)$ in $\mathcal{R}_i$, for value v_k appearing in q', is computed using the CORI algorithm [3, 5]:

$$p(v_k|\mathcal{R}_i) = T_{i,k} \cdot I_k \cdot w_k \tag{2}$$

$$T_{i,k} = \frac{df_{i,k}}{df_{i,k} + 50 + 150 \cdot \frac{cw_i}{\overline{cw}}} \tag{3}$$

$$I_k = \frac{\log\left(\frac{|\mathscr{R}|+0.5}{cf_k}\right)}{\log\left(|\mathscr{R}| + 1.0\right)} \tag{4}$$

where:

w_k is the weight of the term in the query;
$df_{i,k}$ is the number of records in the approximation of $\mathcal{R}_i$ containing value v_k;
cw_i is the number of values in the approximation of $\mathcal{R}_i$;
$\overline{cw}$ is the mean value of all the cw_i;
cf_k is the number of approximated resources containing value v_k;
$|\mathscr{R}|$ is the number of the resources.

In the above formulae, $T_{i,k}$ indicates how many records contain the term v_k in the resource $\mathcal{R}_i$. As cf_k denotes the number of resources in which the term v_k occurs, called *resource frequency*, I_k is defined in terms of cf_k inverse resource frequency: the higher cf_k the smaller I_k, reflecting the intuition that the more a term occurs among the resources the less it is a discriminating term. The belief $p(v_k|\mathcal{R}_i)$ combines these two measures.

Finally, given the query q, all information resources $\mathcal{R}_i \in \mathscr{R}$ are ranked according to their resource relevance value $G(q, \mathcal{R}_i)$, and the top-n are selected as the most relevant ones. This concludes the description of the resource selection phase.

Schema mapping using CORI. We are ready now to describe our method for schema mapping. Let $\mathcal{R}_k \in \mathscr{R}$ be a selected resource. Our task is to find out how to match the attribute-value pairs $A_i = v_i \in q$ (over the target schema) into one or more attribute-value pairs $A_{k_j} = v_i$, where A_{k_j} is an attribute of the schema of the selected resource $\mathcal{R}_k$. The basic idea is as follows. Consider the resource $\mathcal{R}_k$ and the records $r_1, \ldots, r_l$ of the approximation of $\mathcal{R}_k$ $Approx(\mathcal{R}_k)$ (computed by query-based sampling). Each record $r_s \in Approx(\mathcal{R}_k)$ is a set of attribute-value pairs $r_s = \{A_{k_1} = v_{k_1}, \ldots, A_{k_q} = v_{k_q}\}$.

From $Approx(\mathcal{R}_k)$, we make a projection on each attribute, i.e. we build a new set of records for each attribute A_{k_j} of the schema:

$$C_{k,j} = \bigcup_{r_s \in Approx(\mathcal{R}_k)} \{r \mid r = \{A_{k_j} = v_{k_j}\}, A_{k_j} = v_{k_j} \in r_s\} \ .$$

So, each projection $C_{k,1}, \ldots, C_{k,k_q}$ can be seen as a new library, i.e. resource.

Now we apply the resource selection framework for attribute matching: in order to find out whether to match an attribute-value pair $A_i = v_i \in q$ into an attribute-value pair $A_{k_j} = v_i$, we verify whether the resource $C_{k,j}$ has been selected among the top-n relevant resources to the query $\bar{q} = \{A_i = v_i\}$. That is, we build the query $\bar{q} = \{A_i = v_i\}$ and then compute all the goodnesses $G(\bar{q}, C_{k,1}), \ldots, G(\bar{q}, C_{k,k_q})$. If $G(\bar{q}, C_{k,j})$ is the top score, then we map $A_i = v_i$ into the attribute-value pair $A_{k_j} = v_i$. Once we apply the procedure to all $A_i = v_i \in q$, a complex query over the selected source schema $\mathcal{R}_k \in \mathscr{R}$ is obtained and can be submitted to the resource $\mathcal{R}_k$.

4 Experimental Methodology

In this section we describe the data (documents and queries), the evaluation measures, and the experimental setup used to evaluate our approach.

Experimental setup. The schema mapping task involves a target schema, and one source schema with its correspondent resource approximation. The experiments were performed on two different data sets:

– OAI-DC is an Open Archive Initiative collection [3] that contains more than 40,000 scientific documents in XML format. Its schema, used as the source schema, has 21 attributes. As the target schema for this data set we used the NCSTRL system [4], which has 29 attributes.
– NGA is a sampled collection of 864 documents from the National Gallery of Arts, Washington D.C. [5]. The documents are available in a schema manually built from the web site (our source schema), and in a standard schema (our target schema), manually derived from the previous one with simple rules. From now on, the former will be called NGA schema, the latter standard schema. The standard schema contains 12 attributes, while the NGA schema contains 14 attributes.

For each data set, a set of structured queries over the target schema of the form $q = \{A_1 = v_1, \ldots, A_q = v_q\}$ have been submitted to the resource. These queries have been first transformed into simple queries, and then transformed into complex queries by relying on the attribute mappings obtained using the resource selection approach described in Section 3.

For each data set, we perform two sets of experiments: one computing the mappings on the fly when the query is submitted to the resource, and another one by computing the mappings off-line. In the off-line mapping, we use a training set of 10 queries to compute the "best" mapping for each given target attribute. Essentially, for each training query, we compute the set of mappings $A \rightarrow B$, and sum up $A \rightarrow B$'s score. Finally, we rank all the mappings $A \rightarrow B$ in decreasing order according to the final score. For each target attribute A, we select the mapping $A \rightarrow B$ with highest score.

Evaluation Metrics For the evaluation of the effectiveness of the mapping method, we consider two metrics. First, for each query we count how many mappings are correct and report the average value. Second, we evaluate the effectiveness of the query transformation process. That is, given a target query $q = \{A_1 = v_1, \ldots, A_q = v_q\}$, we evaluate whether issuing the simple query $q' = \{v_1, \ldots, v_q\}$ directly to the resource provides better effectiveness than the transformed query. Therefore, different queries are submitted to a resource:

– the optimal query (called, "source query"), which consists in correctly transforming the target query in the source query by using manually built mappings;
– the simple query obtained from the complex target query;
– the query obtained applying on-line mapping;
– the query obtained applying off-line mapping.

For each query submitted to the source schema, the first n results (with n empirically set to 10) are manually evaluated for relevance with respect to standard

[3] http://dublincore.org
[4] RFC 1807 Bibliographic Records Format, http://www.ncstrl.org
[5] http://www.nga.gov

Information Retrieval measurements:

$$Precision = \frac{\#\ RelevantRetrieved}{\#\ TotalRetrieved}$$

$$Recall = \frac{\#\ RelevantRetrieved}{\#\ TotalRelevant}$$

$$F - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

4.1 Experimental Results

The results are reported in Tables 1-3. For each set of queries and each method, the average percentage of mappings correctly found, and the average recall, precision and F-Score are computed.

% mappings	On-line	Off-line
OAI-DC	0.29	0.8
NGA	0.55	1.00

Table 1. % of correct attribute mappings.

Queries	Source	Simple	On-line	Off-line
Avg Precision	0.44	0.32	0.13	0.25
Avg Recall	1.00	0.83	0.34	0.70
Avg F-Score	0.61	0.46	0.19	0.37

Table 2. Effectiveness of schema matching over OAI-DC data set.

Queries	Source	Simple	On-line	Off-line
Avg Precision	0.30	0.29	0.16	0.30
Avg Recall	1.00	0.97	0.60	1.00
Avg F-Score	0.46	0.45	0.25	0.46

Table 3. Effectiveness of schema matching over NGA data set.

Table 1 highlights a considerable difference in effectiveness between the two methods for learning schema mappings. Off-line mapping is clearly better than

on-line mapping, since in the latter effectiveness depends on each query, while in the former it depends on the entire training set.

Applying on-line mapping over OAI-DC is not very effective (Table 2). This may be due to different reasons: a large collection and a small sample (about 1% of the entire collection), which might not give a good resource approximation. Indeed, the OAI-DC resource is such that many query values are distributed in many different and sometimes unjustified source attributes, making the mapping process very difficult. For instance, we found out that the attribute "date" with value "1920" has been mapped into the attribute "source", maybe due to an erroneous metadata compilation by a librarian. Interestingly, better results are obtained applying off-line mapping, which are comparable to the optimal query.

Applying on-line mapping over NGA is not very effective as well (Table 3), even if the gap with the optimum result is lower than in the previous case. In this case, indeed, we have a sample that is about the 10% of the entire collection. Interestingly, applying off-line mapping we obtain the exact mappings for all the target attributes. The experiments were also performed inverting the schemas, using standard as the source schema and NGA as the target schema. The results obtained are exactly the same reported in Table 3.

Note that, in both the data sets used, querying the resources with the simple query (so ignoring the structure of the source schema), we obtain almost the same precision and recall of the original query. This may suggest that query transformation, even if perfect (as e.g. with off-line mapping) is not necessary for querying distributed digital library metadata schemas. However, more investigations should be made on this issue, before drawing concluding results. This case may depend on the particular data set considered, and on the manually constructed queries we have selected (effectiveness of the perfect query almost coincides with the effectiveness of the simple query).

5 Conclusions

In this paper we have proposed the use of CORI resource selection framework to automatize the schema mapping generation phase. This approach allows transformation of a complex query over a target metadata schema into a complex query over the source metadata schema. To the best of our knowledge, a major novelty of this paper is the fact that our approach works directly on queries over the target schema and, thus, no instances of the target schema are required for the learning process.

The results in this paper can be employed for instance in Peer-to-peer networks. These are dynamic scenarios where peers can dynamically join and leave the network, so the system should –for each query– only consider the services which are currently available and relevant to a given query, and transform the query on the fly.

We are currently testing on different data sets to verify whether the good performances of simple query against query transformation is confirmed. We are also investigating the use of bayesian classifiers [28] to determine the schema mappings and compare it againts our resource selection-based approach.

6 Acknowledgements

This work is supported by the ISTI–CNR curiosity driven project "Distributed Search in the Semantic Web".

References

1. J. Berlin and A. Motro. Database schema matching using machine learning with feature selection. In *In Proceedings of the Conf. on Advanced Information Systems Engineering (CAiSE), 2002.*, 2002.
2. Alexander Bilke and Felix Neumann. Schema matching using duplicates. In *Proceedings of the 21st International Conference on Data Engineering (ICDE-05)*, pages –. IEEE Computer Society, 2005.
3. Jamie Callan. Distributed information retrieval. In W.B. Croft, editor, *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, Hingham,MA, USA, 2000.
4. Jamie Callan and Margaret Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.
5. Jamie Callan, Zhihong Lu, and Bruce W. Croft. Searching distributed collections with inference networks. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR-95)*, pages 21–28, Seattle, WA, 1995.
6. Robin Dhamankar, Yoonkyong Lee, AnHai Doan, Alon Halevy, and Pedro Domingos. iMAP: discovering complex semantic matches between database schemas. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 383–394. ACM Press, 2004.
7. H. Do and E. Rahm. Coma - a system for flexible combination of schema matching approaches. In *Proceedings of the Int. Conf. on Very Large Data Bases (VLDB-02), 2002.*, 2002.
8. Anhai Doan, Pedro Domingos, and Alon Halevy. Learning to match the schemas of data sources: A multistrategy approach. *Mach. Learn.*, 50(3):279–301, 2003.
9. AnHai Doan, Pedro Domingos, and Alon Y. Halevy. Reconciling schemas of disparate data sources: a machine-learning approach. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 509–520. ACM Press, 2001.
10. AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12(4):303–319, 2003.
11. David W. Embley, David Jackman, and Li Xu. Multifaceted exploitation of metadata for attribute match discovery in information integration. In *Workshop on Information Integration on the Web*, pages 110–117, 2001.
12. Ronald Fagin, Phokion G. Kolaitis, Reneé Miller, and Lucian Popa. Data exchange: Semantics and query answering. In *Proceedings of the International Conference on Database Theory (ICDT-03)*, number 2572 in Lecture Notes in Computer Science, pages 207–224. Springer Verlag, 2003.
13. Norbert Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems*, 3(17):229–249, 1999.
14. MingChuan Guo and Yong Yu. Mutual enhancement of schema mapping and data mapping. In *In ACM SIGKDD 2004 Workshop on Mining for and from the Semantic Web*, Seattle, 2004.

15. Bin He and Kevin Chen-Chuan Chang. Statistical schema matching across web query interfaces. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 217–228. ACM Press, 2003.
16. R. Ichise, H. Takeda, and S. Honiden. Rule induction for concept hierarchy alignment. In *Proceedings of the Workshop on Ontology Learning at the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001.
17. Jaewoo Kang and Jeffrey F. Naughton. On schema matching with opaque column names and data values. In *Proceedings of the 2003 ACM SIGMOD international conference on on Management of data*, pages 205–216. ACM Press, 2003.
18. Martin S. Lacher and Georg Groh. Facilitating the exchange of explicit knowledge through ontology mappings. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*, pages 305–309. AAAI Press, 2001.
19. J. Madhavan, P. Bernstein, K. Chen, and A. Halevy. Corpus-based schema matching. In *Proceedings of the 21st International Conference on Data Engineering (ICDE-05)*, pages –. IEEE Computer Society, 2005.
20. J. Madhavan, P. Bernstein, K. Chen, A. Halevy, and P. Shenoy. Corpus-based schema matching. In *Workshop on Information Integration on the Web at IJCAI-03*, 2003.
21. Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *Proc. 27th VLDB Conference*, pages 49–58, 2001.
22. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, page 117. IEEE Computer Society, 2002.
23. Henrik Nottelmann and Umberto Straccia. A probabilistic approach to schema matching. In *Proceedings of the 27th European Conference on Information Retrieval Research (ECIR-05)*, Lecture Notes in Computer Science, Santiago de Compostela, Spain, 2005. Springer Verlag.
24. Natalya Fridman Noy and Mark A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455. AAAI Press / The MIT Press, 2000.
25. Lucian Popa, Yannis Velegrakis, Renee J. Miller, Mauricio A. Hernandez, and Ronald Fagin. Translating web data. In *Proceedings of VLDB 2002, Hong Kong SAR, China*, pages 598–609, 2002.
26. Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
27. M. Elena Renda and Umberto Straccia. Web metasearch: Rank vs. score based rank aggregation methods. In *Proc. 18th Annual ACM Symposium on Applied Computing (SAC-03)*, pages 841–846, Melbourne, Florida, USA, 2003. ACM.
28. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
29. Ling Ling Yan, Renée J. Miller, Laura M. Haas, and Ronald Fagin. Data-driven understanding and refinement of schema mappings. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 485–496. ACM Press, 2001.