# Automatic Structured Query Transformation Over Distributed Digital Libraries

M. Elena Renda*
I.I.T. – C.N.R.
and
Scuola Superiore Sant'Anna
I-56100 Pisa, Italy
elena.renda@iit.cnr.it

Umberto Straccia
I.S.T.I. – C.N.R.
I-56100 Pisa, Italy
umberto.straccia@isti.cnr.it

## ABSTRACT

Structured data and complex schemas are becoming the main way to represent the information many Digital Libraries provide, thus impacting the services they offer. When searching information among distributed Digital Libraries with heterogeneous schemas, the structured query with a given schema (the global or target schema) has to be transformed into a query over the schema of the digital library it will be submitted to (the source schema). Schema mappings define the rules for this query transformation. Schema matching is the problem of learning these mappings.

In this paper we address the issue of automatically learning these mappings and transforming a structured query over the target schema into a new structured query over the source schema. We propose a simple and effective schema matching method based on the well known CORI selection algorithm and two ways of applying it. By evaluating the effectiveness of the obtained structured queries we show that the method works well in accessing distributed, heterogeneous digital libraries.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; H.3.4 [**Information Search and Retrieval**]: Systems and Software—*Distributed systems*; H.2.5 [**Database Management**]: Heterogeneous Databases

## General Terms

Algorithms, Experimentation, Measurement, Performance.

## Keywords

Digital Libraries, Structured Queries, Automatic Query Transformation, Automatic Mappings, Schema Matching.

## 1. INTRODUCTION

Digital Libraries (DLs) [14] differ in the kind, quantity and quality of information and services they provide, and in what kind of users they are supposed to be addressed to (see, for instance ACM DL[1], Medline[2], NCSTRL[3]). In particular, they may be extremely heterogeneous under two main aspects:

1. the *topic* of the information they provide. Imagine a wide-area library, with up to thousands of collections available where to search for, dealing with, for instance, Physics, Medicine, Computer Science, and so on; and

2. the *metadata schema* they use to describe the information they provide. Indeed, DLs use different sets of attributes to represent their document content.

As users cannot deal efficiently with the number and the heterogeneity of available DLs, it is becoming increasingly needed that they only deal with a unique system with a unified interface for searching over multiple DLs simultaneously, using one system-wide target metadata schema which is defined independently of the libraries.

For instance, let us assume that:

- we have a *target schema* $T$ with attributes $A_1, \ldots, A_q$ (also called *global* or *mediated schema*);

- a user query $q$ over the target metadata schema $T$ is based on single-valued attributes $A_1, \ldots, A_q$, *i.e.* the 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 target schema $T$ and $v_i$ is a type correct value for $A_i$[4];

- we have $n$ available distributed DLs (called *information resources*) $\mathscr{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$ to search in;

- each resource $\mathcal{R}_i$ has its own metadata schema $S_i$ (the source schema) based on attributes $A_{i_1}, \ldots, A_{i_q}$;

*This work was done when the author was a research assistant at I.S.T.I. – C.N.R.

[1] http://www.acm.org/dl

[2] http://igm.nlm.nih.gov

[3] http://www.ncstrl.org

[4] Note that a query attribute $A_i$ may appear in the query multiple times.

SAC'06, April, 23-27, 2006, Dijon, France.
Copyright 2006 ACM 1-59593-108-2/06/0004 ...$5.00.

- a query $\bar{q}$ over the source schema $S_i$ is a set of attribute-value pairs of the form:

$$\bar{q} = \{A_{i_1} = v_{i_1}, \ldots, A_{i_q} = v_{i_q}\} ,$$

where each $A_{i_j}$ is an attribute of the target schema $S_i$ and $v_{i_j}$ is a type correct value for $A_{i_j}$.

In such a context, three major tasks have to be executed in order to answer the user query $q$ submitted over the target schema $T$: (i) the *Automatic Resource Selection* task (see, e.g. [4, 15]) in order to select a subset of most promising DLs among all the $n$ DLs available, as it is not cost-effective to submit the query to all possible resources; (ii) the *Schema Matching* task (see, e.g. [28]) to reformulate the information need $q$ into a new query $\bar{q}$ over the query language provided by the selected resource(s) using schema mapping rules; and (iii) the *Rank Fusion* (see, e.g. [29]) to merge together the ranked lists obtained by each queried resource.

In this paper we deal with the schema matching problem, *i.e.* with the problem of automatically learning the schema mapping rules for querying heterogeneous distributed DLs.

Schema Matching is based on mappings that usually are 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 we have the metadata target schema $\texttt{T}(\texttt{author}, \texttt{abstract})$ and the resource $\mathcal{R}_i$ with the metadata source schema $\texttt{S}_\texttt{i}(\texttt{creator}, \texttt{description})$. An example of user query over the target schema may be:

$$\begin{aligned} q = \; & \{\texttt{author} = \text{“}Moshe\ Vardi\text{”}, \texttt{abstract} = \text{“}logic\text{”}, \\ & \texttt{abstract} = \text{“}computer\ science\text{”}\} ; \end{aligned}$$

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

$$\begin{aligned} & \texttt{author} \rightarrow \texttt{creator} \\ & \texttt{abstract} \rightarrow \texttt{description} , \end{aligned}$$

the above query $q$ will be rewritten as the query $\bar{q}$ over the source schema:

$$\begin{aligned} \bar{q} = \; & \{\texttt{creator} = \text{“}Moshe\ Vardi\text{”}, \texttt{description} = \text{“}logic\text{”}, \\ & \texttt{description} = \text{“}computer\ science\text{”}\} . \end{aligned}$$

We propose a simple and effective method to automatically learn schema mappings in such a scenario, which is based on the well known CORI resource selection method [7]. Indeed, similarly to the resource selection problem, where we have to automatically identify the most relevant libraries with respect to a given query, in the schema matching problem we have to identify, for each target attribute, the most relevant source attribute with respect to a given structured query. That is, given (i) an attribute-value pair $A_k = v_k$, with $A_k$ being an attribute of the target schema $T$, and (ii) a resource $\mathcal{R}_i$ with the source schema $S_i$, we want to identify among all the attributes $A_{i_j} \in S_i$ the most relevant one to map $A_k$ to.

While most of the schema matching methods presented in the literature need an a priori schema learning process, thus requiring instances of the target schema, a major feature and, to the best of our knowledge, the novelty of our approach is that the method we propose can directly be applied to queries over the target schema without requiring instances of it. This is often the case when we deal with queries over distributed DLs, as training data is missing.

The structure of the paper is the following: Section 2 presents an overview of the schema matching problem and its main application fields. Section 3 introduces our formal framework for schema matching applied to query transformation. In Section 4 we report the evaluation of the proposed approach on different data sets. Section 5 concludes.

## 2. RELATED WORK

Matching is a fundamental task in the manipulation of structured information; it consists in taking two schemas/ontologies, each consisting of a set of entities as input (*e.g.*, tables, XML elements, classes, properties, predicates, metadata) and producing as output a mapping, *i.e.* a semantic relationships (*e.g.*, equivalence, subsumption) between elements of the two given schemas [8, 10, 23, 27].

With the proliferation of DLs over the Web, the development of automated tools for schema matching is of particular importance to automatize distributed information retrieval. Indeed, matching has a central role in many well-known application domains, such as Semantic Web, schema integration, Electronic Commerce (E-Commerce), Ontology integration, web-oriented data integration, XML mapping, Catalog matching, etc.

Manually performing Schema Matching, as it is usually done when searching the Web, is time-consuming, and expensive. Moreover, the number and the complexity of Web information resources to integrate grows enormously, thus making it difficult to handle the Schema Matching process.

Matching has been addressed by many researchers in two related areas: the matching problem for *ontologies* and the matching problem for *database* schemas. *Ontology Matching* differs from *Schema Matching* substantially for two reasons ([25]): presence of explicit semantics and knowledge models. Indeed, while ontologies are logical systems that themselves incorporate semantics (intuitive or formal), database schemas often do not provide explicit semantics for their data. Furthermore, ontology data models are richer than schema data models. The approaches proposed for Ontology Matching [12, 20, 21, 26] try to exploit knowledge explicitly encoded in the ontologies, while Schema Matching approaches try to guess the meaning encoded in the schemas.

However, even considering these differences, these two areas can be seen as closely related. Indeed, schemas can be considered as, *e.g.*, simple ontologies with restricted relationship types; on the other hand, the techniques applied in schema matching can be applied to ontology matching as well, taking care of the hierarchies.

All the Schema Matching approaches proposed in the literature [3, 9, 18, 22, 24] involve the definition of mappings among schemas and, possibly, of a global integrated schema. 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 [1, 11]. In particular, applying machine learning techniques requires instances from both the target schema 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. An extensive comparison can be found in [28].

Typical application for schema matching are *Schema Integration*, *i.e.* the problem of constructing a global view, given a set of independently developed schemas [30], “message translation” for E-Commerce portals, and also the Semantic Web [2, 13], where matching can be used for mapping messages between autonomous agents.

In this paper we consider another scenario for Schema Matching: given a user query, written with the target schema $T$ of the library $R_m$, automatically learn for each attribute $T_i \in T$ of the query the most promising attribute $S_j$ of the source schema $R_n$, in order to submit the query over $R_m$ to the resource $R_n$. [16] has proved that automatic structured queries may significantly improve the search process over structured DLs.

# 3. QUERY TRANSFORMATION

In this Section we describe our approach for solving the schema matching problem in order to automatically learn schema mappings and transform queries over heterogeneous DLs. It is based on a reformulation of the CORI resource selection framework [7]. In the following, in order to make the paper self-contained, we first introduce the automatic resource selection task and describe how it is performed using the CORI method; after that we fit the CORI resource selection method into our context.

## Automatic Resource Selection Using CORI

The *Automatic Resource Selection* task is becoming a crucial step when searching over a large number of distributed, heterogeneous resources. Indeed, resource selection improves search efficiency, reducing the costs of distributed search (search time, network bandwidth, computation, and so on), and can also improve search effectiveness, since it is supposed that after the selection the resources with no relevant documents are not queried, thus gaining in time and documents relevance.

Some approaches presented in the literature rely on short resource descriptions [17, 31], a sort of content summary that usually includes the terms that appear in the resource and their document frequency; furthermore, it may include other simple information, such as the total number of documents (the resource dimension). Web resources and their corresponding search servers are usually non-cooperative, while all the approaches relying on the resource description require their cooperation in order to obtain content summary or information about the documents they index, such as term occurrence statistics. For this reason, *query-based sampling* techniques have been proposed for deriving content description approximations [5, 19]. These approaches use queries (called *probe queries*) to retrieve and download a relatively small set of documents (called the *resource sample*), representative of the topic covered by the resource; the resource sample is then used to build the content summary (the *description* or *approximation* of the resource) by extracting term occurrence statistics. In [6] it has been demonstrated that resource samples provide statistics quite representative of the statistics of the full resource. The resource description is then used 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 to a given user query.

CORI is one of the methods used in automatic resource selection to compute the resource score with respect to the query. In the following we describe an adapted version of the CORI resource selection method in case documents are represented via metadata records.

Consider the user query $q = \{A_1 = v_1, \ldots, A_q = v_q\}$ over the target schema $T$. At first, we unfold the complex query $q$ into a simple query $q'$ where the attributes $A_i$ have been removed, *i.e.* $q' = \{v_1, ..., v_q\}$ . For each resource $\mathcal{R}_i \in \mathscr{R}$, we compute the *resource score* $G(q, \mathcal{R}_i)$ (also called *goodness*), representing the relevance of resource $\mathcal{R}_i$ to the query $q$, as follow:

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

where $|q'|$ is the number of values in the simple query $q'$. The *belief* $p(v_k|\mathcal{R}_i)$ is computed for each value $v_k \in q'$ using the CORI algorithm [4, 7]:

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

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

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

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 the 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 the value $v_k$; |
| $\lvert\mathscr{R}\rvert$ | 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 $\mathcal{R}_i$, while $I_k$, defined in terms of the *resource frequency* $cf_k$, is the *inverse resource frequency*: the higher $cf_k$ the smaller $I_k$, *i.e.*, 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. Informally, a resource is more relevant if its approximation, computed by query-based sampling, contains many terms related to the original query, but if a query term occurs in many resources, this term is not a good one to discriminate between relevant and not relevant resources.

Finally, all the information resources $\mathcal{R}_i \in \mathscr{R}$ are ranked according to their resource goodness value $G(q, \mathcal{R}_i)$, and the top-*k* are selected as the most relevant ones.

## Schema Matching Using CORI

Let $q = \{A_1 = v_1, \ldots, A_q = v_q\}$ be a user query over the metadata target schema $T$ and $\mathcal{R}_k \in \mathscr{R}$ a relevant selected resource, with metadata source schema $S_k$. Our task is to find out how to map each attribute-value pair $A_i = v_i \in q$ into one or more attribute-value pairs $A_{k_j} = v_i$, where $A_{k_j} \in S_k$, using CORI.

The basic and simple idea is the following. Consider a query attribute-value pair $A_i = v_i \in q$ and let $\mathcal{R}_k \in \mathscr{R}$ be a selected resource where to search into. Let $A_{k_1}, ..., A_{k_q} \in S_k$ be all the attributes of the metadata schema of $\mathcal{R}_k$. Our idea is to map the attribute $A_i$ to the attribute $A_{k_j}$ if the value $v_i$ is relevant to the collection of all the values the attribute $A_{k_j}$ takes in the resource $\mathcal{R}_k$. Essentially, for each of the attributes $A_{k_j} \in S_k$, we make a collection $C_{k,j}$ of the values the attribute $A_{k_j}$ takes in the resource $\mathcal{R}_k$ and then ask CORI which of these collections are most relevant to the query value $v_i$. If $C_{k,j}$ is among the answers of CORI , then we build the mapping $A_i \rightarrow A_{k_j}$.

More formally, 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:

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

The basic idea is that each projection $C_{k,1}, \ldots, C_{k,k_q}$ can be seen as a new collection, and we apply CORI to select which of these is the most relevant for each attribute-value pairs $A_i = v_i$ of the query $q$.

By using each projection $C_{k,j}$ as a resource, we can apply the resource selection framework for attribute matching: in order to find out whether to match a target attribute-value pair $A_i = v_i \in q$ into a source 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 $q^* = \{A_i = v_i\}$. That is, we build the query $q^* = \{A_i = v_i\}$ and then compute all the goodnesses $G(q^*, C_{k,1}), \ldots, G(q^*, C_{k,k_q})$. If $G(q^*, C_{k,j})$ 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 $\bar{q} = \{A_{k_1} = v_1, \ldots, A_{k_q} = v_q\}$ over the selected source schema $\mathcal{R}_k \in \mathscr{R}$ is obtained and can be submitted to the resource $\mathcal{R}_k$.

# 4. EXPERIMENTS

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 corresponding resource approximation. The experiments were performed on three different data sets:

- the OAI-DC collection[5], which is an Open Archive Initiative collection containing more than $40,000$ scientific documents in XML format;
- a set of $800$ computer science documents in the XML OAI-RFC $1807$ format gathered from the NCSTRL collection (Networked Computer Science Technical Reference Library[6]). From now on we will call this data set OAI-RFC; and
- the NGA collection, a sampled set of $864$ records from the National Gallery of Arts, Washington D.C. [7].

In the OAI-DC data set the documents are available in an XML schema (our source schema) with 21 attributes. As the target schema for this data set we used the NCSTRL bibliographic schema (the RFC 1807 Bibliographic Records Format), which has 29 attributes. The OAI-DC resource approximation was built by sending $187$ randomly generated probe queries and the number of records collected was $391$.

In the OAI-RFC data set the documents are available in an XML schema (our source schema) with 29 attributes. As the target schema for this data set we used the OAI-DC schema. The OAI-RFC resource approximation was built by sending $116$ randomly generated probe queries and the number of records collected was $150$.

In the NGA data set the documents are available in a schema manually built from the web site, and in a standard 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 has 12 attributes, while the NGA schema has 14 attributes. We performed the experiments once considering the NGA schema as our source schema and the standard schema as our target schema, and once by inverting their roles. The resource approximation of NGA was built by sending $213$ randomly generated probe queries and the number of records collected was $90$, while the resource approximation of the collection using the standard schema was built by sending $125$ randomly generated probe queries and the number of the unique records collected was $158$.

For each data set, a set of manually generated 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 transformed into complex queries $q_i^c = \{A_{i_1} = v_{i_1}, \ldots, A_{i_q} = v_{i_q}\}$, by relying on the attribute mappings obtained using the resource selection method CORI, as described in Section 3.

These mappings were computed in two different ways: one applying an on-line mapping and one with an off-line mapping method. In the former case the schema matching process is performed on the fly when the query is submitted to the resource; consequently, it is possible to obtain a mapping in the query $q_i$ for a given attribute $A_m$ different from the one obtained in the query $q_j$ for the same attribute[8].

In the off-line mapping, a training set of 10 queries have been used to compute off-line, a-priori of the searching process, the "best" mapping (if any) for each given target attribute. Essentially, for each training query, we compute the set of mappings $A \rightarrow B_i$, and sum up $A \rightarrow B_i$'s scores. Finally, we rank all the mappings $A \rightarrow B_i$ in decreasing order according to the final score. For each target attribute $A$, we select the mapping $A \rightarrow B_i$ with highest score. In this way, the mapping used for each attribute is the same for all the queries at search time.

Actually, for both on-line and off-line mapping, we performed a set of experiments where we selected once the best mapping found for each attribute ($k = 1$), once the first 2 ($k = 2$), and so on. The experimental results have demonstrated that the $k = 1$ selection (*i.e.*, select the best mapping) is the most effective one.

**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 (Table 1).

Second, we evaluate the effectiveness of the query transformation process, *i.e.*, given a target query $q = \{A_1 = v_1, \ldots, A_q = v_q\}$ we evaluate the effectiveness of issuing the transformed complex query $\bar{q} = \{A_{i_1} = v_1, \ldots, A_{i_q} = v_q\}$, obtained by applying the mapping rules found.

To evaluate the results we compare them with the optimal results, obtained by submitting, for each query, the correspondent optimal query, *i.e.* the target query correctly transformed in the source query by using manually built mappings. Actually in querying the resource we proceed in two different ways:

1. we submit each complex query $q_i^c = \{A_{i_1} = v_{i_1}, \ldots, A_{i_q} = v_{i_q}\}$ and retrieve the first $n$ results from the obtained score-based list;
2. we submit, for each complex query $q_i^c = \{A_{i_1} = v_{i_1}, \ldots, A_{i_q} = v_{i_q}\}$, $|q|$ queries of the form $q_{i_j}^c = \{A_{i_j} = v_{i_j}\}$ to the resource, *i.e.*, we search each attribute-value pair separately, and then we linearly combine the $|q|$ result lists obtained.

In both cases, the results obtained are the same. This means that our results are not affected by the search engine query mechanism used to retrieve the results from the resources.

**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.

Concerning the percentage of correct mappings (Table 1), suppose that a user query $q_i$ has 4 attribute-value pairs, and the match-

[5] http://www.openarchives.org

[6] http://www.ncstrl.org

[7] http://www.nga.gov

[8] Note that it is also possible that for a given attribute there is no mapping found.

| % mappings | On-line | Off-line |
|---|---|---|
| OAI-DC | 0.29 | 0.80 |
| OAI-RFC | 0.49 | 0.55 |
| NGA | 0.55 | 1.00 |

**Table 1: Percentage of correct attribute mappings.**

| **OAI-DC** | | |
|---|---|---|
| Queries | On-line | Off-line |
| Avg Precision | 29.55 | 56.82 |
| Avg Recall | 34.00 | 70.00 |
| Avg F-Score | 31.15 | 60.66 |
| **NGA** | | |
| Queries | On-line | Off-line |
| Avg Precision | 53.33 | 100.00 |
| Avg Recall | 60.00 | 100.00 |
| Avg F-Score | 54.35 | 100.00 |
| **OAI-RFC** | | |
| Queries | On-line | Off-line |
| Avg Precision | 75.00 | 90.00 |
| Avg Recall | 70.00 | 82.00 |
| Avg F-Score | 75.76 | 87.88 |

**Table 2: Effectiveness of schema matching over the 3 data sets, tested using the sample.**

ing method returns the exact match for 2 of them; the correct mapping percentage for $q_i$ is then $0.50$. Note that if, for instance, the target attribute `year` is mapped into the source attribute `datestamp` by means of the mapping rule found, but the right (manually identified) source attribute would be `date`, we consider this as a wrong mapping, even if `date` and `datestamp` are semantically the same. In Table 1 the percentage of mappings correctly found is computed as the average of the correct mapping percentage found for all the queries.

Table 1 highlights a considerable difference in effectiveness between the two variants of the method we propose for learning schema mappings. Off-line mapping is clearly better than on-line mapping, and this was predictable 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. By applying off-line mapping the results obtained for precision and recall are highly better.

Even if on-line mapping performs better if applied over NGA and OAI-RFC, the off-line mapping performance is always better (Table 2). The experiments with the NGA data set 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 as those reported in Table 2 for NGA.

Since the sample influences the quality of mappings (for the probe query randomness, and for the number and quality of the records retrieved in the sampling phase) we decided to perform another set of experiments where the mappings were computed considering the entire collection instead of the sample.

| **OAI-DC** | | |
|---|---|---|
| Queries | On-line | Off-line |
| Avg Precision | 63.64 | 52.27 |
| Avg Recall | 72.00 | 67.00 |
| Avg F-Score | 65.57 | 55.74 |
| **NGA** | | |
| Queries | On-line | Off-line |
| Avg Precision | 83.33 | 100.00 |
| Avg Recall | 70.00 | 100.00 |
| Avg F-Score | 80.43 | 100.00 |
| **OAI-RFC** | | |
| Queries | On-line | Off-line |
| Avg Precision | 90.00 | 95.00 |
| Avg Recall | 82.00 | 87.00 |
| Avg F-Score | 87.88 | 93.94 |

**Table 3: Effectiveness of schema matching over over the 3 data sets, tested using the entire collection.**

The results are reported in Table 3, and comparing them with those obtained on the sample (Table 2) we can assert that: though on-line and off-line mappings performed on the entire collection are more effective than the one computed considering only the sample, they enormously slow down the matching phase, due to the large number of records to check for finding attribute mappings., and they are expensive, both in terms of computation time (and user time, in case of on-line mapping.)

We can conclude that the best way to apply this method for automatically learning the attribute mappings for query transformation when querying heterogeneous resources is the off-line mapping.

## 5. CONCLUSIONS

In this paper we have proposed the use of the CORI resource selection framework to automate the generation of schema mappings and the transformation of queries over heterogeneous Digital Libraries. This approach allows transformation of a complex query over a target metadata schema into a complex query over a 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 unlike other approaches it does not require instances of the target schema for the learning process.

We have chosen the CORI resource selection framework because, to the best of our knowledge, it is one of the most promising. As further research, we plan to evaluate other resource selection frameworks applied in schema matching, and compare the results with classical methods used in query transformation and schema matching.

The results in this paper will be employed in peer-to-peer networks, dynamic scenarios where peers can dynamically join and leave the network. In particular, in hierarchical peer-to-peer networks the subset of peers acting as directory services could apply our method to dynamically transform the query depending on the source schema of the resource(s) selected as relevant for the given query.

## 6. ACKNOWLEDGEMENTS

This work was supported by the ISTI–CNR curiosity driven project “Distributed Search in the Semantic Web”.

## 7. REFERENCES

[1] J. Berlin and A. Motro. Database schema matching using machine learning with feature selection. In *Proceedings of the 14th Conference on Advanced Information Systems Engineering (CAiSE-02)*, 2002.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *The Scientific American*, 284(5):34–43, 2001.

[3] A. Bilke and F. Neumann. Schema matching using duplicates. In *Proceedings of the 21st International Conference on Data Engineering (ICDE-05)*, pages 69–80. IEEE Computer Society, 2005.

[4] J. Callan. Distributed information retrieval. In W. Croft, editor, *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, Hingham, MA, USA, 2000.

[5] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.

[6] J. Callan, M. Connell, and A. Du. Automatic discovery of language models for text databases. In *Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD-99)*, pages 479–490, Philadelphia, Pennsylvania, USA, 1999. ACM Press.

[7] J. Callan, Z. Lu, and B. 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 (SIGIR-95)*, pages 21–28, Seattle, WA, 1995.

[8] R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. iMAP: discovering complex semantic matches between database schemas. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 383–394. ACM Press, 2004.

[9] H. Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proceedings of the International Conference on Very Large Data Bases (VLDB-02).*, pages 610–621, 2002.

[10] A. Doan, P. Domingos, and A. Y. Halevy. Reconciling schemas of disparate data sources: a machine-learning approach. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 509–520, Santa Barbara, California, US, 2001. ACM Press.

[11] A. Doan, P. Domingos, and A. Y. Halevy. Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning*, 50(3):279–301, 2003.

[12] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12(4):303–319, 2003.

[13] A. Doan, J. Madhavan, P. Domingos, and A. Y. Halevy. Learning to map between ontologies on the semantic web. In *Proceedings of the 11th International Conference on World Wide Web (WWW-02)*, pages 662–673. ACM Press, 2002.

[14] E. A. Fox and G. Marchionini. Digital libraries: Introduction. *Communications of the ACM*, 44(5):30–32, 2001.

[15] N. Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems*, 3(17):229–249, 1999.

[16] M. Goncalves, E. Fox, A. Krowne, P. Calado, A. F. Laender, A. da Silva, and B.Ribeiro-Neto. The effectiveness of automatically structured queries in digital libraries. In *Proceedings of the 4th ACM/IEEE-CS joint conference on digital libraries (JCDL-04)*, pages 98–107, New York, NY, USA, 2004. ACM Press.

[17] L. Gravano, H. Garcia-Molina, and A. Tomasic. GlOSS: Text-source discovery over the internet. *ACM Transactions on Database Systems*, 24(2):229264, 1999.

[18] B. He and K. Chang. Statistical schema matching across web query interfaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 217–228. ACM Press, 2003.

[19] P. G. Ipeirotis and L. Gravano. Distributed search over the hidden web: Hierarchical database sampling and selection. In *Proceedings of the twenty-eighth International Conference on Very Large Data Bases (VLDB-02)*, pages 394–405, 2002.

[20] Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.

[21] M. S. Lacher and G. 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.

[22] 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 57–68. IEEE Computer Society, 2005.

[23] T. Milo and S. Zohar. Using schema matching to simplify heterogeneous data translation. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 122–133. Morgan Kaufmann Publishers Inc., 1998.

[24] H. Nottelmann and U. Straccia. A probabilistic approach to schema matching. In *Proceedings of the 27th European Conference on Information Retrieval Research (ECIR-05)*, LNCS, Santiago de Compostela, Spain, 2005. Springer Verlag.

[25] N. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *In Knowledge and Information Systems*, 6(4):428–440, 2004.

[26] N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the 17th National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455. AAAI Press / The MIT Press, 2000.

[27] L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernandez, and R. Fagin. Translating web data. In *Proceedings of VLDB 2002, Hong Kong SAR, China*, pages 598–609, 2002.

[28] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[29] M. E. Renda and U. Straccia. Web metasearch: Rank vs. score based rank aggregation methods. In *Proceedings 18th Annual ACM Symposium on Applied Computing (SAC-03)*, pages 841–846, Melbourne, Florida, USA, 2003. ACM.

[30] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, 1990.

[31] J. Xu and J. Callan. Effective retrieval with distributed collections. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–120, 1998.