Information Retrieval and Machine Learning for Probabilistic Schema Matching*

Henrik Nottelmann
University of Duisburg-Essen
Lotharstrasse 65
47048 Duisburg, Germany
nottelmann@uni-duisburg.de

Umberto Straccia
ISTI-CNR
Via G. Moruzzi 1
56124 Pisa, Italy
straccia@isti.cnr.it

ABSTRACT

Schema matching is the problem of finding correspondences (mapping rules, e.g. logical formulae) between heterogeneous schemas. This paper presents a probabilistic framework, called sPLMap, for automatically learning schema mapping rules. Similar to LSD, different techniques, mostly from the IR field, are combined. Our approach, however, is also able to give a probabilistic interpretation of the prediction weights of the candidates, and to select the rule set with highest matching probability.

Categories and Subject Descriptors: H.2.1 [Database Management]: Logical Design - Schema and subschema

General Terms: Theory, Experimentation

Keywords: sPLMap, schema matching, predicate logics, probability theory

1. INTRODUCTION

Combining heterogeneous data from different sources, i.e. transforming data structured under one schema into data structured under a different schema, is an old but emerging problem in the "information age" (e.g. for federations of existing digital libraries like the ACM DL or CiteSeer). This problem is currently under investigation in the context of *information integration* [3] and *data exchange* [2]. When schemas change frequently over time, or a large amount of schemas is involved, automatic mapping generation (called *schema matching*) is mandatory.

This paper describes the sPLMap framework (probabilistic, logic-based mapping between schemas) [4, 5] for automatic schema matching, which is founded on information retrieval and machine learning techniques. From well-known approaches like LSD [1], it borrows the idea of combining several specialized components (called "classifiers") for finding the best mapping, e.g. based on attribute name comparison or comparing properties of the underlying data instances. The major improvements over LSD are the support for data types (e.g. text, names, different date formats) in the matching process, and the usage of probability theory for estimating the degree of correctness of a learned rule. Thus, it provides a sound theoretical justification for its (optimum) selection, and a measure for the quality of a learned schema mapping.

In the following, a *schema* $\mathbf{R} = \langle R_1, \ldots, R_n \rangle$ is a tuple of binary relations R_i containing document identifiers and attribute values (belonging to a data type, e.g. "Text", "Name", "Year" or "DateISO8601"). This structure corresponds to a linear list of multi-valued *schema attributes*. The instance of an attribute R_i is also denoted by R_i, and, similarly, $\mathbf{R}$ is used for the schema and the corresponding *schema instance*.

A *schema mapping rule* has the form $S_i \rightarrow T_j$. Formally, it specifies that pairs of document identifier and attribute values have to be copied from source instance S_i into target instance T_j. Let Σ_j be a set of rules with common rule head T_j. Then, $\widehat{T}_j = \bigcup_{i_k} S_{i_k}$ denotes the result of applying all rules from Σ_j onto the source instance $\mathbf{S}$; thus, $\widehat{T}_j$ is an instance over the target attribute T_j. $\widehat{\mathbf{T}} = \langle \widehat{T}_1, \ldots, \widehat{T}_t \rangle$ denotes the instance derived by applying Σ onto $\mathbf{S}$. Typically, the set Σ of mapping rules on which $\widehat{T}_j$ and $\widehat{\mathbf{T}}$ depend is clear from the context and left out.

2. LEARNING SCHEMA MAPPINGS

Consider a target schema $\mathbf{T} = \langle T_1, \ldots, T_t \rangle$ and a source schema $\mathbf{S} = \langle S_1, \ldots, S_s \rangle$ together with two instances (which do not necessarily describe the same documents). The goal is to find the "best" set of mapping rules Σ which maximizes the probability $Pr(\Sigma, \mathbf{T}, \mathbf{S})$ that the tuples in $\mathbf{T}$ and in the mapping result $\widehat{\mathbf{T}}$ are 'similar', that is, given a random tuple in $\mathbf{T}$, estimate the probability that it is a tuple in $\widehat{\mathbf{T}}$ and vice-versa.

Each Σ can be partitioned into t sets $\Sigma_1, \ldots, \Sigma_t$ (where any Σ_j contains only rules with rule head T_j). As the rules in these sets operate independently onto different target attributes, we obtain:

$$Pr(\Sigma, \mathbf{T}, \mathbf{S}) = \prod_{j=1}^{t} Pr(\Sigma_j, \mathbf{T}, \mathbf{S}) \ .$$

Now, $Pr(\Sigma_j, \mathbf{T}, \mathbf{S})$ is estimated as the probability that a tuple in $\widehat{T}_j$ is also in T_j, and vice-versa:

$$Pr(\Sigma_j, \mathbf{T}, \mathbf{S}) = Pr(T_j | \widehat{T}_j) \cdot Pr(\widehat{T}_j | T_j) = Pr(\widehat{T}_j | T_j)^2 \cdot \frac{|T_j|}{|\widehat{T}_j|} \ .$$

Unfortunately, there are exponentially many possible sets Σ_j. For computational simplification, we assume that S_{i_1} and S_{i_2} are dis-

*This work is supported in part by the DFG (grant BIB47 DOuv 02-01, project "Pepper") and in part by ISTI-CNR (project "Distributed Search in the Semantic Web").

CIKM'05, October 31–November 5, 2005, Bremen, Germany.
ACM 1-59593-140-6/05/0010.

Schema-based		Content-based	
CL_N	same name	CL_L	value overlap
CL_S	same name stem	CL_{kNN}	kNN
CL_D	same data type	CL_B	Naive Bayes
		CL_{KL}	KL-distance

Table 1: Classifiers

joint for $i_1 \neq i_2$. This leads us to the approximation:

$$Pr(\widehat{T}_j|T_j) \approx \sum Pr(S_{i_k}|T_j) \ .$$

Thus, in order to compute $Pr(\Sigma_j, \mathbf{T}, \mathbf{S})$, the main task is to compute the $O(s \cdot t)$ probabilities $Pr(S_i|T_j)$. Similar to LSD [1], we combine different classifiers $CL_1, \ldots, CL_n$ (see table 1 for classifiers we use). Each classifier outputs a conditional probability $Pr(S_i|T_j, CL_k)$ as an approximation of $Pr(S_i|T_j)$. The final probability $Pr(S_i|T_j)$ is computed by combining the classifier predictions $Pr(S_i|T_j, CL_k)$ in a weighted sum:

$$Pr(S_i|T_j) \approx \sum_{k=1}^{n} Pr(S_i|T_j, CL_k) \cdot Pr(CL_k) \ .$$

In a first step, each classifier CL_k computes an initial weight $w(S_i, T_j, CL_k)$, which is then normalized (due to heterogeneous scales) into the probability $Pr(S_i|T_j, CL_k) = f(w(S_i, T_j, CL_k))$. In a first normalization step, an arbitrary function can be used, e.g. linear or logistic functions. A second normalization ensures that the final values for $Pr(S_i|T_j, CL_k)$ and for $Pr(T_j|S_i, CL_k)$ are in $[0, 1]$.

Finally, the probability $Pr(CL_k)$ describes the probability that we rely on the judgment of classifier CL_k. We simply set $Pr(CL_k) = \frac{1}{n}$ (i.e., we average over all classifiers). Alternatively, one could aim at learning these probabilities for each classifiers individually.

3. EVALUATION

This section briefly describes some experimental results; more results can be found in [4, 5]. We use a 2.382 document BibTeX collection (BIBDB) with a large overlap in attribute names, and an Open Archive collection of the Library of Congress (LOC) with 1.337 entries, available in MARC 21 and Dublin Core. Both collections are split randomly into three sub-collections of approximately the same size (two for learning, one for evaluation). Classifiers from table 1 are used in isolation as well as in combination. As all LOC attributes use the same data type "Text", and do not share common names, only the content-oriented classifiers are used for this collection.

The results in table 2 use the typical IR measures precision, recall and F-measure. For BIBDB, the combination of all classifiers yields 82% of precision and recall. According to the F-measure, the best schema classifier are CL_N and CL_S and the worst is CL_D, while the best content classifier is CL_L and the worst is CL_{KL}. For LOC, the combination of all classifiers yields 64% of precision and 22% of recall only. The best content classifier is CL_{kNN} (which slightly outperforms the combination) and the worst is CL_{KL}.

There is no general best classifier. Which classifier performs well always depends on the concrete schemas (and their instances). The experiments show for BIBDB (and other similar collections we cannot report here due to space limitations) satisfactory results for the combination of all classifiers, while the performance is worse for the difficult LOC collection (where the DC attributes often contain text which equals the concatenation of strings from multiple MARC 21 attributes). Typically, the name-based classifiers provide good precision and rather low recall, while using data types

(a) BIBDB

	Prec	*Rec*	*F*
CL_N	1.0000	0.6364	0.7778
CL_S	1.0000	0.6364	0.7778
CL_D	0.2632	0.9091	0.4082
CL_{kNN}	0.5455	0.5455	0.5455
CL_B	0.5000	0.4545	0.4762
CL_L	0.8750	0.6364	0.6368
CL_{KL}	0.2353	0.7273	0.3556
all schema	0.4348	0.9091	0.5882
all content	0.4211	0.7273	0.5333
all	0.8182	0.8182	0.8182

(b) LOC

	Prec	*Rec*	*F*
CL_{kNN}	0.6667	0.2439	0.3571
CL_B	0.4375	0.1707	0.2456
CL_L	0.8000	0.1951	0.3137
CL_{KL}	0.1714	0.2927	0.2162
all content	0.6429	0.2195	0.3273

Table 2: Effectiveness evaluation

perform worse. Content-oriented classifiers proved to be very useful for schema matching (w. r. t. both precision and recall), which is important in general settings where schema names alone are not sufficient to find matching attributes. Here, kNN and Naive Bayes provide high precision. CL_{KL} yields low precision but can improve recall when combined with other classifiers. The experiments further showed that combining classifiers can clearly increase the matching quality.

4. CONCLUSION AND OUTLOOK

We have presented sPLMap, a Probabilistic, Logic-based formal framework for the important schema matching problem. The peculiarity of our approach is that it combines neatly machine learning, information retrieval and heuristic techniques for learning a set of mapping rules. The framework has already been extended towards uncertain rules [4, 5]. In future, we plan to develop and evaluate additional classifiers CL_K, and learn the probabilities $Pr(CL_k)$. This framework can easily be extended towards more complex rules, which e.g. combine the content of several attributes, and to new application areas like ontologies.

5. REFERENCES

[1] A. Doan, P. Domingos, and A. Y. Halevy. Reconciling schemas of disparate data sources: a machine-learning approach. In *ACM SIGMOD international conference on Management of data*, 2001.

[2] R. Fagin, P. G. Kolaitis, R. Miller, and L. Popa. Data exchange: Semantics and query answering. In *International Conference on Database Theory (ICDT)*, 2003.

[3] M. Lenzerini. Data integration: a theoretical perspective. In *ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS)*, 2002.

[4] H. Nottelmann and U. Straccia. A probabilistic approach to schema matching. Technical Report 2004-TR-60, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy, 2004.

[5] H. Nottelmann and U. Straccia. sPLMap: A probabilistic approach to schema matching. In *European Conference on Information Retrieval Research (ECIR)*, 2005.