

SEMANTIC MATCHING

Across Heterogeneous Data Sources

Discovering semantic correspondences is the key to semantic integration of data sources and ultimately to data integration across disparate databases.

As our ability to build information systems improves, so does the need to integrate the ones we build. For instance, we need cooperation among massively distributed information systems for homeland security. It must be possible to retrieve information about a suspected individual from many systems maintained nationwide, even worldwide, by any number of organizations, including intelligence agencies, police departments, motor vehicle departments, and all kinds of transportation providers and authorities. We also need a

[B y H u i m i n Z h a o]

unified master patient index that integrates (legally and ethically) numerous health care systems and allows authorized care providers to access the medical records of all patients [1]. The growth of the Internet, especially Web services, amplifies the need for semantic interoperability across heterogeneous data sources. Related data sources accessible via different Web services create new requirements and opportunities for data integration, both within and across organizations, in many domains.

The systems that must be integrated are typically heterogeneous in a variety of ways, including operating systems, data models, database management systems (DBMSs), application programming languages, structural formats, and data semantics. Many techniques help bridge the systematic and structural gaps across heterogeneous systems; examples are heterogeneous DBMSs, connectivity middleware (such as ODBC, OLE DB, and JDBC), and emerging XML-based Web services technology [5]. However, semantic alignment across systems is a resource-consuming process that demands automated support. A critical step in semantic integration is determining semantic correspondences across the underlying heterogeneous data sources. Here, I explore techniques that are potentially useful for facilitating intersystem semantic matching.

Semantic correspondences across heterogeneous data sources include schema-level correspondence and instance-level correspondence. Schema-level correspondence consists of tables in different data sources that describe the same real-world entity type and attributes in different data sources that describe the same property of some entity type. Instance-level correspondences consist of records in different data sources yet that represent the same real-world entity. The problem of determining schema-level correspondences is called schema matching and interschema relationship identification [2, 12]. The process of determining instance-level correspondence is called record matching (or linkage) and instance (or entity) identification [4, 10, 11].

Consider an example of two security-related data-

bases (outlined in Table 1 and Table 2) owned by different organizations. They contain several common attributes and overlapping records, though they also include discrepancies and data errors. Restricting the scope of relevant data to just the two tables, data analysts responsible for integrating the two databases must identify the corresponding attributes (such as `Suspect.FirstNm` and `Criminal.FName`) and corre-

| FirstNm | LastNm | Gender | Hair | Eyes | Height | Weight |
|---------|--------|--------|-------|-------|--------|--------|
| Andrew | Keafer | Male | Black | Black | 5'8" | 160 |
| Lillian | Lee | Female | Black | Black | 5'2" | 130 |
| Carole | Smith | Female | Blond | Blue | 6'3" | 310 |

Table 1. Sample entries in table `Suspect` of database A.

sponding records (such as the first record of `Suspect` and the first record of `Criminal`), making it possible to link or integrate the tables.

Since real-world data sources are often quite large—with possibly hundreds of tables, thousands of attributes, and millions of records—manually identifying their correspondences tends to be prohibitively expensive. Human analysts need automated or semi-automated tools (based on expert rules or on learning techniques) to help discover them. In a rule-based approach, domain experts provide the decision rules for identifying them [6]. In a learning-based approach, decision rules are learned through machine learning techniques [2, 7, 9–12]. Since the rule-based approach involves time-consuming knowledge acquisition to elicit domain knowledge from human experts, the learning-based approach is preferred, as it bypasses the bottleneck of knowledge acquisition.

Unsupervised learning (or cluster analysis) and supervised learning (or classification) techniques automate the discovery of semantic correspondences. Cluster analysis techniques are generally more suited to identifying schema-level correspondences, and classification techniques are generally more suited to detecting instance-level correspondences. Cluster analysis performed automatically by a tool recommends rough groups of similar examples in a data set but needs to be redone whenever the data set changes. Classification “learns” a decision model based on a training sample to make specific predictions—whether two records match—on new data. The num-

SINCE REAL-WORLD DATA SOURCES ARE OFTEN QUITE LARGE—with possibly hundreds of tables, thousands of attributes, and millions of records—manually identifying their correspondences tends to be prohibitively expensive.

ber of tables and attributes is usually much smaller than the number of records in a data source. While some amount of follow-up manual review by human data analysts of clustering results is affordable for schema matching, this review is less likely to be cost-effective for record matching. Moreover, schemas are relatively more stable than instance records. There is no need to repeat schema matching unless many new

| FName | LName | Sex | HairColor | EyeColor | Height | Weight |
|---------|-------|-----|-----------|----------|--------|--------|
| Andy | Kefer | 1 | BLK | BLK | 173 | 73 |
| Lillian | Li | 2 | BLK | BLK | 157 | 58 |
| Carol | Smith | 2 | BLD | BLU | 190 | 140 |

data sources are to be integrated dynamically, but record matching must be performed whenever the data in the underlying data sources is updated.

SCHEMA ELEMENTS AND INSTANCES

Clustering tools compare schema elements based on a variety of characteristics, including names, documents, specifications, and data and usage patterns. As tables and attributes are named to reflect their meanings, string-matching methods and linguistic tools (such as thesauri) are used to measure the similarities among table names and attribute names. Descriptions of tables and attributes in design documents can be compared through document-similarity measures developed in the information-retrieval field. Attributes representing similar concepts tend to be modeled through similar specifications (such as data type, length, and constraints). Similar attributes also tend to exhibit similar data patterns (such as length and formation, as in proportions of digits, letters, and white spaces, and number of distinct values and percentage of missing values). Similar schema elements are also used to, say, update frequency and number of users and user groups. Clustering tools use these specifications, data patterns, and usage patterns to compare schema elements.

Due to the potential for confusion, selection of the characteristics for comparing schema elements in a particular application must be done carefully. Schema elements are frequently named with ad hoc abbreviations rather than with regular words. Design documents are often outdated, incomplete, incorrect, ambiguous, or simply not available. Semantically similar concepts can be modeled using different structures; for example, “gender” can be defined as a numeric attribute in one data source and as a character in another data source. Data patterns are correlated more with structures than with semantics. Usage data may not be maintained in legacy systems. Other

semantics and business rules may simply reside in human minds or may be deeply embedded in hard code. It is therefore necessary to utilize multiple types of clues and also involve multiple domain experts in the process to capture their domain knowledge.

Schema-level correspondences are the basis for comparing records across heterogeneous data sources. A pair of records in corresponding tables can be compared based on a set of corresponding attributes to determine whether or not the records match when the tables do not share a common key. If two corresponding attributes are accurately recorded following the same format, they can be compared literally. However, real-world data sources often involve discrepancies.

Semantically corresponding attributes often have different formats in different data sources; for example, (414)2296524 and 1-414-229-6524 both refer to the same phone number. The same attribute can be measured on different scales (such as the metric kilogram vs. the U.S. pound) in different data sources. Most operational databases include wrong data, spelling errors, and different abbreviations. Human names are often misspelled, mistaken for similar-sounding names (such as Keafer and Keefer), or substituted with nicknames (such as Andy and Andrew).

The result is that different transformation procedures and approximate matching functions must be used to standardize the format and measure the similarity between corresponding attributes. There are many approximate string-matching methods, some of which (such as edit distance) account for spelling errors (such as insertions, deletions, substitutions, and transpositions of characters) and others (such as Soundex) for phonetic errors. Special-purpose methods standardize and compare particular types of attributes (such as human names and addresses). And special translators (such as the two-letter abbreviation for each U.S. state) help resolve coding differences across databases.

The result is that different transformation procedures and approximate matching functions must be used to standardize the format and measure the similarity between corresponding attributes. There are many approximate string-matching methods, some of which (such as edit distance) account for spelling errors (such as insertions, deletions, substitutions, and transpositions of characters) and others (such as Soundex) for phonetic errors. Special-purpose methods standardize and compare particular types of attributes (such as human names and addresses). And special translators (such as the two-letter abbreviation for each U.S. state) help resolve coding differences across databases.

LEARNING TECHNIQUES

Figure 1 classifies several widely used learning techniques. Cluster analysis techniques group the examples in a data set into groups (called clusters) of similar examples. Because the groups from the data are previously unknown, cluster analysis is characterized as “unsupervised” learning. When applied to schema matching, schema elements are classified into clusters of similar elements based on their characteristics (such as names, documents, specifications,

Table 2. Sample entries in table Criminal of database B.

data patterns, and usage patterns) described earlier. These groups of similar schema elements are then presented to domain experts for further evaluation.

Cluster analysis is supported by many statistical and neural network techniques [12]. Statistical clustering techniques may be hierarchical or nonhierarchical. A nonhierarchical one (such as K-means) requires that users specify the desired number of clusters. A hierarchical one clusters examples on a series of similarity levels, from very fine to very coarse partitions. Hierarchical methods can be agglomerative or divisive. Agglomerative techniques start from the finest partition (in which each individual example is a cluster) and successively merge smaller clusters into larger clusters. Divisive methods start from the coarsest partition (in which all the schema elements are in a single cluster) then successively divide big clusters into smaller clusters. Kohonen's Self-Organizing Map (SOM) is an unsupervised neural network that projects high-dimensional data onto a low-dimensional (usually 2D) space. SOM is especially good at visualizing the proximity among schema elements. Since no clustering method has been found to be the single best choice, several methods must be used together to achieve an optimal solution.

A classification technique is used to build a general prediction model (called a classifier) that can be used to help predict the value of a dependent variable (called class) based on a set of explanatory variables. Because the classifier is derived from a set of training examples whose classes are given, classification is characterized as "supervised" learning. The learned classifier can then be used to predict the classes of other examples. When applied to record matching, a pair of records from different data sources is classified into one of two classes—match and non-match—based on their similarity scores on corresponding attributes. Domain experts should manually classify some record pairs for training the classifier.

Classification is supported by many statistical, machine learning, and neural network methods [11]. Four widely used statistical methods are Naive

Bayes, Fellegi and Sunter's record linkage theory [4], logistic regression, and k -nearest neighbor. Naive Bayes estimates the odds ratio (a record pair being match vs. non-match) under the assumption that the explanatory variables are conditionally independent. Fellegi and Sunter's record linkage theory extends Naive Bayes specifically for the record linkage problem, allowing a record pair to be classified into one of three classes: match, non-match, and unclassified. Logistic regression finds a linear boundary—a weighted sum of the explanatory variables—to separate the two classes—match and non-match. K -nearest neighbor simply memorizes the training examples and classifies each new example into the

majority class of the k closest training examples. Machine learning techniques generate decision tables, trees, and rules. Two widely used techniques are C5 and CART. Back propagation is a widely used neural network technique for classification. Neural networks are highly interconnected, with an input layer, an output layer, and zero or more intermediate layers;

they successively adjust the weights of the connections among the nodes on neighboring layers during training.

Methods are also available for combining multiple classifiers to further improve classification accuracy; examples include bagging, boosting, cascading, and stacking. Bagging and boosting train multiple classifiers of the same type—with homogeneous base classifiers—making the final prediction based on the votes of the base classifiers. In bagging, the base classifiers are trained independently using different training data sets and given equal weight in the voting. In boosting, base classifiers are learned sequentially (each new classifier focuses more on the examples classified incorrectly by previous classifiers) and are weighted according to their accuracy. Cascading and stacking combine classifiers of different types (with heterogeneous base classifiers). Cascading combines classifiers vertically, with the output of one base classifier used as an additional input variable for the next base classifier. Stacking combines classifiers horizontally, with the output of several base classifiers used as input variables for a high-level classifier responsible for making the final classification decision.

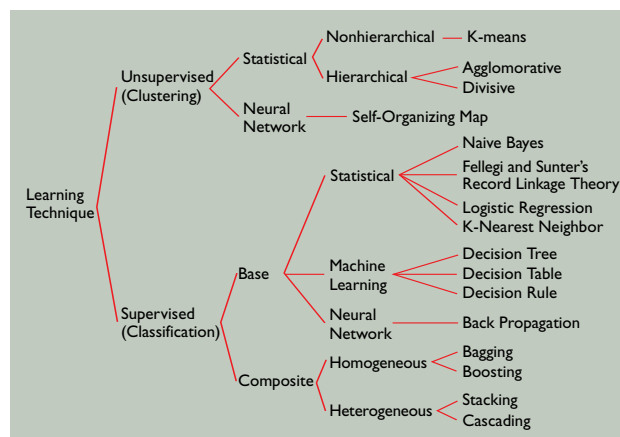
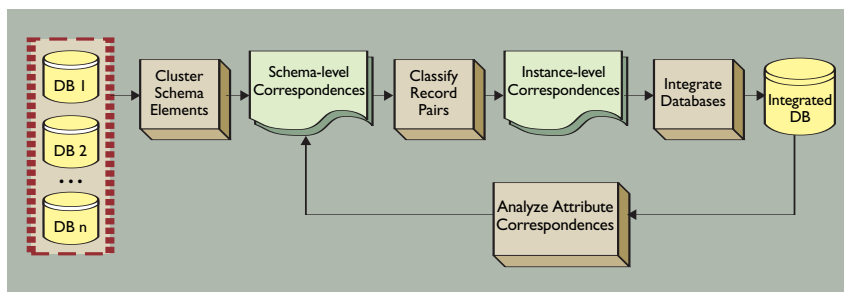


Figure 1. Widely used learning methods.

CLASSIFIERS MAY NOT BE ABLE TO CLASSIFY ALL RECORD PAIRS with sufficient accuracy, leaving some difficult cases for analysts to review manually.

COMPREHENSIVE PROCEDURE

Semantic correspondences on the schema level and the instance level are related. Schema-level correspondences provide the necessary basis for comparing records. Given identification of corresponding records, attribute correspondences can be evaluated



more accurately using statistical analysis techniques [3]. It is therefore productive to combine the two techniques into a comprehensive procedure, so the accuracy of identified correspondences on the two levels is gradually improved [8].

Figure 2 outlines a general procedure for semantic matching across heterogeneous data sources, starting with clustering schema elements. Schema-level correspondences suggested by cluster analysis are reviewed and verified by domain experts and then used to determine corresponding records using classification techniques. After some corresponding records are identified, data from different data sources is linked or integrated together and further analyzed using statistical analysis techniques. Semantically related attributes tend to be highly correlated and can be identified through correlation analysis. Regression analysis can further determine the actual relationship (such as scaling discrepancy) among correlated attributes. Correspondences among categorical attributes can be analyzed using a more general statistical-dependence measure (such as mutual information). For example, a normalized mutual information index between any two attributes is zero if the attributes are statistically independent and 100% if the attributes are one-to-one transformations of each other. If such statistical analysis reveals any new findings, record matching can

be redone at the human analyst's discretion. Similarly, if new corresponding records are identified, statistical analysis of attribute correspondences can be performed again. This procedure is repeated until no further improvement in the results is obtainable.

Human analysts should keep in mind that the procedure is not fully automated and that human intervention might have to be applied at each step.

Cluster analysis is highly empirical, requiring careful evaluation of its results. Classifiers may not be able to classify all record pairs with sufficient accuracy, leaving some difficult cases for analysts to review manually. Highly correlated attributes detected by statistical analysis techniques may describe related but not identical properties about some entity type, thus requiring that analysts verify whether the correlated attributes

are indeed corresponding in light of domain knowledge. Tools help human analysts but never totally replace them.

Figure 2. A general procedure for semantic matching across heterogeneous data sources.

MATCHING SECURITY DATABASES

Now consider how the procedure and its various techniques might be applied in matching the two security-related databases in Table 1 and Table 2. First, the attributes must be clustered, as the scope is restricted to just the two corresponding tables. The attribute names can be compared using a string-matching method (such as edit distance) to account for different abbreviations (such as FirstNm and FName) and a thesaurus to account for synonyms (such as Gender and Sex). If descriptions of the attributes are available, they can be compared using a string- or document-similarity measure. Data patterns (such as summary statistics like average, standard deviation, and range) can be computed for each attribute. Specifications and usage patterns can also be used. If there are too many such characteristics for cluster analysis, a dimensionality reduction tech-


nique (such as principal component analysis, or PCA) can be used first to reduce the number of input variables. PCA produces a few linear combinations of the original variables (called “principal components”) that may roughly represent the original data set. Cluster-analysis techniques (such as K-means, hierarchical clustering, and SOM) can then be applied to cluster the attributes based on these principal components.

After some corresponding attributes are identified, various classification techniques can be used by the human analyst to identify corresponding records. If these records have a common key (such as Social Security number or driver’s license number), training examples are easily generated using the key. Otherwise, the analyst needs to manually classify record pairs for training classifiers. The human analyst can also build different transformation and matching procedures to compare corresponding attributes. Attributes measured on different scales (such as Suspect.Weight measured in U.S. pounds and Criminal.Weight measured in metric kilograms) require re-scaling. Categorical attributes coded differently (such as Suspect.Gender using “male” and “female” and Criminal.Sex using 1 and 2, respectively) require special translators. Human names can be compared by combining several matching methods (such as Soundex for matching similar-sounding names like Keafer and Keefer), nickname dictionary for matching different nicknames (such as Andy and Andrew), and edit distance for handling spelling errors (such as Carol and Carole).

Corresponding records can be integrated into a single data set so statistical analysis can be used to further analyze the relationships among attributes. Correlation analysis can be used to find highly correlated attributes. Regression can be used to discover transformation formulae among corresponding attributes (such as $\text{Suspect.Weight} = 2.2 \times \text{Criminal.Height}$). Mutual information can be used to detect categorical attributes coded differently in the tables (such as Suspect.Gender and Criminal.Sex). These differences can be analyzed more rigorously than during the cluster analysis discussed earlier. Note that some related but different attributes may be correlated to some extent; for example, Weight and Height may be somewhat correlated, and the analyst should evaluate the analysis results, crossing out such spurious correspondences. This semantic matching procedure can be repeated until the analyst is satisfied with the results.

CONCLUSION

Many techniques are available for determining semantic correspondences across heterogeneous data

sources—a key step in the semantic integration of the sources. After more than two decades of extensive research in heterogeneous database integration, it is time to harvest some of the resulting procedures, techniques, and tools. Data analysts can combine these techniques, incorporate them into comprehensive tools, and validate and improve them in real-world, large-scale data-integration applications. In the meantime, we still must identify the related difficulties and how well the techniques perform in real applications. Real human insights gained from practice are crucial for assuring the relevance of theoretical semantic matching research. 

REFERENCES

1. Bell, G. and Sethi, A. Matching records in a national medical patient index. *Commun. ACM* 44, 9 (Sept. 2001), 83–88.
2. Doan, A. and Domingos, P. Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning* 50, 3 (Mar. 2003), 279–301.
3. Fan, W., Lu, H., Madnick, S., and Cheung, D. DIRECT: A system for mining data value conversion rules from disparate sources. *Decision Support Systems* 34, 1 (Dec. 2002), 19–39.
4. Fellegi, P. and Sunter, A. A theory of record linkage. *Journal of the American Statistical Association* 64, 328 (Dec. 1969), 1183–1210.
5. Hansen, M., Madnick, S., and Siegel, M. Data integration using Web services. In *Lecture Notes in Computer Science 2590*. Springer, 2003, 165–182.
6. Hernandez, M. and Stolfo, S. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery* 2, 1 (Jan. 1998), 9–37.
7. Li, W. and Clifton, C. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data & Knowledge Engineering* 33, 1 (Apr. 2000), 49–84.
8. Ram, S. and Zhao, H. Detecting both schema-level and instance-level correspondences for the integration of e-catalogs. In *Proceedings of the Workshop on Information Technology and Systems* (New Orleans, Dec. 15–16, 2001), 187–192.
9. Tejada, S., Knoblock, C., and Minton, S. Learning object identification rules for information integration. *Information Systems* 26, 8 (Dec. 2001), 607–633.
10. Verykios, V., Elmagarmid, A., and Houstis, E. Automating the approximate record-matching process. *Information Sciences* 126, 1–4 (July 2000), 83–98.
11. Zhao, H. and Ram, S. Entity identification for heterogeneous database integration: A multiple classifier system approach and empirical evaluation. *Information Systems* 30, 2 (Apr. 2005), 119–132.
12. Zhao, H. and Ram, S. Clustering schema elements for semantic integration of heterogeneous data sources. *Journal of Database Management* 15, 4 (Jan.–Mar. 2004), 88–106.

HUIMIN ZHAO (hzhao@uwm.edu) is an assistant professor of management information systems in the Sheldon B. Lubar School of Business Administration at the University of Wisconsin - Milwaukee.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright of Communications of the ACM is the property of Association for Computing Machinery and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.