Chapter 6

RELATED WORK

In this chapter we review works that relate to our representation-matching solution and discuss in detail how our solution advances the state of the art.

- First, we review formal semantics that have been developed for representation matching, as well as proposed notions of similarity.

- Second, we survey the vast body of matching solutions that have been developed in both the database and AI communities. We compare these solutions to ours from several perspectives, ans show how our solution provides a unifying framework for most current solutions.

- Third, our work has made contributions to several learning issues, such as multi-strategy learning, learning with structured data, and relaxation labeling. Hence, we also review works related to such learning scenarios.

- Finally, we discuss works in other knowledge-intensive domains (e.g., information extraction and solving crossword puzzles) which bear interesting resemblances to representation matching.

6.1 Formal Semantics and Notions of Similarity

Several works have addressed the issue of formal semantics for representation matching. In [BC86] the authors introduced the notion of *integration assertions* which relate the elements in two schemas (and therefore are essentially semantic mappings). Given two schemas S and T, an integration assertion has the form $e = f$, where e and f are expressions defined over the elements of S and T, respectively. The meaning of such an integration assertion is that there exist interpretations I_S and I_T (for S and T, respectively) that map e and f into the same concept in the universe.

In [MHDB02] the authors introduce more expressive forms of semantic mappings. In their framework a mapping is of the form $e \mathsf{op} f$, where e and f are defined as above, and the operator op is well defined with respect to the output types of e and f. For example, if both expressions have relations as output types, then op can be $=$ and $\subseteq$. If e outputs a constant and f outputs a unary relation, then op can be $\in$ [1].

In the above work the authors also show that some times one needs a helper representation to relate two expressions in S and T. For example if e and f refer to the students in Seattle and San Francisco, respectively, then they are disjoint sets and hence cannot be related directly to each other.

[1] In [MHDB02] the authors use the term *formula* to refer to a semantic mapping (as in our framework), and use *a mapping* to refer to the set of semantic mappings between the two given representations (and optionally a helper representation).

In this case, we need to relate both of them to the concept of students in a helper model. In the same work the authors also identify and study several important properties of mappings such as query answerability, mapping inference, and mapping composition.

Our formal semantics framework (described in Chapter 2) builds on previous works [MHDB02, BC86], but extends them in several important ways.

- First, we always use a helper representation (as introduced by [MHDB02]). This representation is the user domain representation $\mathcal{U}$ defined in Chapter 2. This simplifies the conceptual framework.

- Second, we introduce the notion of similarity distance between the elements (and expressions) in $\mathcal{U}$. We assume the user can define an *arbitrary* measure of similarity over concepts in the domain representation $\mathcal{U}$. This is in marked contrast to previous works, which either do not consider any similarity notion, or only very restricted forms of it (see the discussion on notions of similarity below). In our work, we contend that a similarity notion is a fundamental and integral part of the user's conceptualization of the domain, and hence must be given explicitly. The introduction of similarity notion provides a formal explanation for the working of representation matching algorithms: they attempt to approximate true similarity values using the syntactic clues (as discussed in Section 2.3.1 of Chapter 2).

- Finally, previous works define an expression, such as *e*, to be built from the elements of a representation, such as *S*, and a set of operators. The operators are *well defined over representation S*. This could be problematic if *S* and *T* use different representation languages. For example, suppose *S* is a relational representation and *T* is an XML one. Now consider a mapping that equates a nested XML element f in *T* with an expression *e* in *S*. Obviously, *e* must use some XML operators to construct an output type that is the same as the output type of f. However, it would be difficult to give well-defined semantics to such XML operators over the relational representation *S*. To avoid this problem, we describe all operators involved (in both *S* and *T*) as having semantics over the user domain representation $\mathcal{U}$ (see Section 2.3.2 of Chapter 2 for more details).

Notions of Similarity: Several works have considered the notion of similarity between concepts. The similarity measure in [RHS01] is based on the κ (Kappa) statistics, and can be thought of as being defined over the joint probability distribution of the concepts involved. In [Lin98] the authors propose an information-theoretic notion of similarity that is also based on the joint distribution. However, these works argue for a single best universal similarity measure, whereas we argue for the opposite. Furthermore, our solutions (e.g., GLUE) actually allow handling multiple application-dependent similarity measures. There have been many works on notions of similarity in machine learning, case-based reasoning, and cognitive psychology. For a survey of semantic similarity discussed in many such works, see Section 8.5 of [MS99].

6.2 Representation-Matching Algorithms

Matching solutions have been developed primarily in the database and AI communities. In this section we review and compare these solutions to ours from several perspectives.

6.2.1 Rule- versus Learner-based Approaches

Rule-based Solutions: The vast majority of current solutions employ hand-crafted rules to match representations. Works in this approach include [MZ98, PSU98, CA99, MWJ, MBR01, MMGR02] in databases and [Cha00, MFRW00, NM00, MWJ] in AI.

In general, hand-crafted rules exploit schema information such as element names, data types, structures, and number of subelements. A broad variety of rules have been considered. For example, the TranScm system [MZ98] employs rules such as "two elements match if they have the same name (allowing synonyms) and the same number of subelements". The DIKE system [PSU98, PSTU99, PTU00] computes the similarity between two representation elements based on the similarity of the characteristics of the elements and the similarity of related elements. The ARTEMIS and the related MOMIS [CA99, BCVB01] system compute the similarity of representation elements as a weighted sum of the similarities of name, data type, and substructure. The CUPID system [MBR01] employs rules that categorize elements based on names, data types, and domains. Rules therefore tend to be domain-independent, but can be tailored to fit a certain domain, and domain-specific rules can also be crafted.

Learner-based Solutions: Recently, several works have employed machine learning techniques to perform matching. Works in this direction include [LC00, CHR97, BM01, BM02, NHT+02] in databases and [PE95, NM01, RHS01, LG01] in AI.

Current learner-based solutions have considered a variety of learning techniques. However, any specific solution typically employs only a single learning technique (e.g., neural networks or Naive Bayes). Learning techniques considered exploit both schema and data information. For example, the SemInt system [LC94, LCL00, LC00] uses a neural-network learning approach. It matches schema elements based on field specifications (e.g, data types, scale, the existence of constraints) and statistics of data content (e.g., maximum, minimum, average, and variance).

The DELTA system [CHR97] associates with each schema element a text string that consists of the element name and all other meta-data on the element, then matches elements based on the similarity of the text strings. DELTA uses information-retrieval similarity measures, like the *Name Learner* in LSD. The ILA system [PE95] matches the schemas of two sources by analyzing the description of objects that are found in both sources. The Autoplex and Automatch systems [BM01, BM02] use a Naive Bayes learning approach that exploits data instances to match elements. The HICAL system [RHS01] exploits the data instances in the overlap between the two taxonomies to infer mappings. The system described in [LG01] computes the similarity between two taxonomic nodes based on their signature TF/IDF vectors, which are computed from the data instances.

Rahm and Bernstein [RB01] provide the most recent survey on matching solutions, and describe some of the above works in detail. The survey in [BLN86] examines earlier works on matching which used mostly rule-based techniques. Both surveys consider works that have been developed in the database community.

Comparison of the Two Approaches: Each of the above two approaches – rule-based and learner-based – has its advantages and disadvantages. Rule-based techniques are relatively inexpensive. They do not require training as in learner-based techniques. Furthermore, they typically operate only on schemas (not on data instances), and hence are fairly fast. They can work very well in certain types of applications. For example, in ontology versioning a frequent task is to match two

consecutive versions of an ontology [NM02]. The consecutive versions tend to differ little from each other, and hence are very amenable to rule-based techniques, as [NM02] shows. Finally, rules can provide a quick and concise method to capture valuable user knowledge about the domain. For example, the user can write regular expressions that encode times or phone numbers, or quickly compile a collection of county names or zip codes that help recognize those types of entities. As another example, in the course-listing domain, the user can write the following rule: "use regular expressions to recognize elements about times, then match the first time element with start-time and the second element with end-time". Notice that learning techniques would have difficulties being applied to these scenarios. They either cannot learn the above rules, or can do so only with abundant training data or with the right representations for training examples.

On the other hand, rule-based techniques also have major disadvantages. First, they cannot exploit data information effectively, even though the data can encode a wealth of information (e.g., value format, distribution, frequently occurring words, and so on) that would greatly aid the matching process. Second, they cannot exploit previous matching efforts, such as the initial mappings that the user manually created in the case of the LSD system (Chapter 3). Thus, in a sense, systems that rely solely on rule-based techniques have difficulties learning from the past, to improve over time. Finally, rule-based techniques have serious problems with schema elements for which no effective hand-crafted rules can be found. For example, it is not clear how one can hand craft rules that distinguish between movie description and user comments on the movies, both being long textual paragraphs.

In a sense, learner-based techniques are complementary to rule-based ones. They can exploit data information and past matching activities. They excel at matching elements for which hand-crafted rules are difficult to obtain. However, they can be more time-consuming than rule-based techniques, requiring an additional training phase, and taking more time processing data and schema information. They also have difficulties learning certain types of knowledge (e.g., times, zipcodes, county names, as mentioned above). Furthermore, current learner-based approaches employ only a single learner, and thus have limited accuracy and applicability. For example, the neural-network technique employed by SemInt does not handle textual elements very well, and the objects-in-the-overlap technique of ILA makes it unsuitable to the common case where sources do not share any object.

The Combination of Both Approaches in Our Solution: The complementary nature of rule- and learner-based techniques suggest that an effective matching solution should employ both – each whenever it is deemed effective. Our work in this dissertation offers a technique to do so. The multistrategy framework – introduced in LSD and subsequently extended in COMAP and GLUE – employs multiple base learners to make matching predictions, then combines their predictions using a meta-learner. While the majority of base learners that we have described employ learning techniques, it is clear that, in general, base learners can also employ hand-crafted rules. Our solution employs a meta-learning technique (stacking in Chapter 3) to automatically find out the effectiveness of each base learner in different situations. The multistategy framework therefore represents a significant step toward an effective and unifying matching solution.

6.2.2 Exploiting Multiple Types of Information

Many works in representation matching exploit multiple types of information, such as names, data types, integrity constraints, attribute cardinality, and so on. However, they employ a single strategy for this purpose. For example, the SemInt system [LC94, LCL00, LC00] employs neural networks, the Autoplex system [BM01] employs Naive Bayes classification techniques, and the DELTA system [CHR97] lumps all information about an element into a single long piece of text, then matches the pieces using information retrieval techniques.

Some works have considered several different matching strategies, based on the heuristic that the combination of multiple strategies may improve matching accuracy. The hybrid system described in [CHR97], for example, combines the predictions of the SemInt and DELTA system. However, these works combine strategies in a hardwired fashion, thus making it extremely difficult to add new strategies. Several recent works [CA99, BCVB01, DR02] solve the above problem by using schemes such as weighted sum to combine predictions coming from different matching strategies. The weights employed in such solutions must be hand-tuned, based on the specific application context.

This dissertation advances the state of the art on exploiting multiple types of information in several important aspects. First, we *bring this issue to the forefront* of representation matching, with our work on LSD. We clearly show that there are many different types of information available, and that a matching solution must exploit all of them to maximize matching accuracy.

Second, we *consider a much broader range* of information types than the previous works. Specifically, we advocate building a solution that can exploit both schema and data information, domain integrity constraints, heuristic knowledge, previous matching activities, user feedback, and other types of user knowledge about the matching application (e.g., similarity measure).

Third, we make the case that *there is no one-size-fit-all technique*: each type of information should be exploited using an appropriate strategy, be it Naive Bayes, neural network, decision tree, hand-crafted rule, or recognizer. This point has not been articulated in previous works on representation matching.

Fourth, we *introduce multistrategy learning* as a technique that can automatically select the weights that are used to combine multiple strategies. Thus, we provide a solution to the problem of manually tuning the weights (which is both tedious and inaccurate). However, multistrategy learning is not limited to just the use of weights. It also raises the possibility of employing more sophisticated techniques to combine strategies, such as decision trees or Bayesian networks.

Finally, we show for the first time that *the same multistrategy approach can also be carried over to complex matching* (Chapter 4).

6.2.3 Incorporating Domain Constraints and Heuristics

It was recognized early on that domain integrity constraints and heuristics provide valuable information for matching purposes. Hence, almost all the works we have mentioned exploit some forms of this type of knowledge.

In most works, integrity constraints have been used to match representation elements *locally*. For example, many works match two elements if they participate in similar constraints (among other things). The main problem with this scheme is that it cannot exploit "global" constraints and heuristics that relate the matching of *multiple* elements (e.g., "at most one element matches house-

address"). To address this problem, in this dissertation we have advocated moving the handling of constraints to *after* the matchers. This way, the constraint handling framework can exploit "global" constraints and is highly extensible to new types of constraints.

While integrity constraints are *domain-specific* information (e.g., house-id is a key for house listings), heuristic knowledge makes *general* statements about how the matching of elements relate to each other. A well-known example of a heuristic is "two nodes match if their neighbors also match", variations of which have been exploited in many systems (e.g., [MZ98, MBR01, MMGR02, NM01]). The common scheme is to *iteratively* change the mapping of a node based on those of its neighbors. The iteration is carried out one or twice, or all the way until some convergence criterion is reached.

Our GLUE work provides a solution to exploit a broad range of heuristic information, including those heuristics that have been commonly used in the matching literature. The solution builds on a well-founded probabilistic interpretation, and treats domain integrity constraints as well as heuristic knowledge in a uniform fashion.

6.2.4 Handling User Feedback

Most existing works have focused on developing automatic matching algorithms. They either ignore the issue of user interaction, or treat it as an afterthought. The typical assumption is that whenever a system cannot decide (e.g., between multiple matching alternatives), then it asks the user [MZ98].

The exceptions are several recent works in ontology matching [Cha00, MFRW00, NM00]. These works have powerful features that treat user feedback as an integral part of the matching process and allow for efficient user interaction. For example, the system in [NM00] frequently solicits user feedback on its matching decisions (e.g., confirm or reject the decisions), then makes subsequent decisions based on the feedback.

The Clio system [MHH00, YMHF01, PVH$^+$02] focuses on very fine-grained mappings, which are for example SQL or XQuery expressions that can be immediately executed to translate data from one representation to another. Clio makes two important contributions. First, it recognizes that creating such fine-grained mappings entails making decisions that require user input. Deciding if inner join or outer join should be used is an example of such decisions. Hence, like the previous works in ontology matching that we just described, it also brings the user to the center of the matching process. Second, it realizes that efficient interaction with the user is crucial to the success of matching. Hence, it develops techniques to minimize the amount of interaction required.

The key innovation we made regarding user feedback is that we treat such feedback as temporary domain constraints and heuristics. Thus, we allow users to specify as little or as much feedback as necessary. Our framework also allows users to iteratively interact with the matching system in an efficient manner (e.g., by rerunning the relaxation labeler as many times as necessary).

An important issue that Clio has touched on, and that we have not considered, is finding out how to minimize user interaction – asking them only what is absolutely necessary – and yet make the most out of such interaction. We shall return to this topic when we discuss future directions in the next chapter (Chapter 7).

6.2.5 1-1 and Complex Matching

The vast majority of current works focus only on finding 1-1 semantic mappings. Several works (e.g., [MZ98]) deal with complex matching in the sense that such matchings are hard-coded into rules. The rules are systematically tried on the elements of given representations, and when such a rule fires, the system returns the complex mapping encoded in the rule.

As mentioned earlier, the Clio system [MHH00, YMHF01, PVH+02] creates complex mappings for relational and XML data. To create a complex mapping for a representation element, Clio assumes that the "right" attributes and formula have been given (either by the user, by data mining techniques, or by systems such as LSD). It then focuses on finding the "right" relationship between the attributes (see Chapter 4 for more detail on "right" attributes, formula, and relationships).

In a sense, our work (with the COMAP system) is complementary to Clio in that we find the "right" attributes and formula, assuming the "right" relationship is given. We show in Chapter 4 that our current framework can be extended to address the question of finding the "right" relationship. We believe that a complete and practical system to deal with complex mappings can be developed by combining the multi-searcher architecture and the learning/statistical techniques of COMAP with the powerful facilities for user interaction and for developing fine-grained mappings of Clio.

6.2.6 Generic vs. Application-Specific Solutions

A recent interesting trend covers both ends of the representation matching spectrum. At one end, there have been several works that focus on developing very specialized, application-specific matching solutions. The rationale for this is that representation matching is so difficult, that we should specialize our solution to exploit application-specific features. An example of such works is [NM02], which focuses on matching multiple versions of the same ontology. As mentioned, since consecutive versions tend to differ little from each other, solutions that utilize simple rules can be developed that achieve very high matching accuracy.

At the other end, several works have advocated building generic matching solutions (e.g., [RB01, DR02] and this dissertation), mostly because representation matching is a fundamental step in numerous data management applications. In the foreseeable future, it is likely that there will be a need for, and we shall continue to see, works in both directions.

6.2.7 Further Related Work

The works [Ber03, PB02] discuss model management and schema matching in that context. The work [RD00] discusses data cleaning and schema matching. Several recent works [RRSM01, RMR00, RS01, SRLS02] discuss the issue of building large-scale data integration systems in detail and the crucial role of schema matching in this process. The work [SR01] discusses the impact of XML on data sharing, in particular schema matching and object matching. The work [EJX01] discusses a schema matching approach that is similar to LSD, but using a different set of base learners and a simple averaging method to combine the base learners' predictions.

6.3 Related Work in Learning

We now briefly survey works that are related to learning issues in this dissertation.

Combining Multiple Learners: Multi-strategy learning has been researched extensively [MT94], and applied to several other domains (e.g., information extraction [Fre98], solving crossword puzzles [KSL+99], and identifying phrase structure in NLP [PR00]). In our context, our main innovations are the three-level architecture (base learners, meta-learner and prediction combiner) that allows learning from both schema and data information, and the use of integrity constraints to further refine the learner.

Learning with Structured Data: Yi and Sundaresan [YS00] describe a classifier for XML documents. However, their method applies only to documents that share the same DTD, which is not the case in our domain.

Relaxation Labeling for Learning to Label Interrelated Instances: This technique has been employed successfully to similar matching problems in computer vision, natural language processing, and hypertext classification [HZ83, Pad98, CDI98]. Our work on relaxation labeling is most similar to the work on hypertext classification of [CDI98]. The key difference is that we consider more expressive types of constraints and a broader notion of neighborhood. As a consequence, the optimization techniques of [CDI98] do not work efficiently for our context. To solve this problem, we develop new optimization techniques that are shown empirically to be accurate and extremely fast (see Section 5.3.3). These techniques are general and hence should also be useful for relaxation labeling in other contexts.

Exploiting Domain Constraints: Incorporating domain constraints into the learners has been considered in several works (e.g., [DR96]), but most works consider only certain types of learners and constraints. In contrast, our framework allows arbitrary constraints (as long as they can be verified using the schema and data), and works with any type of learner. This is made possible by using the constraints during the matching phase, to restrict the learner predictions, instead of the usual approach of using constraints during the training phase, to restrict the search space of learned hypotheses.

6.4 Related Work in Knowledge-Intensive Domains

Representation matching requires making *multiple interrelated inferences*, by combining a *broad variety* of relatively *shallow* knowledge types. In recent years, several other domains that fit the above description have also been studied. Notable domains are information extraction (e.g., [Fre98]), solving crossword puzzles [KSL+99], and identifying phrase structure in NLP [PR00]. What is remarkable about these studies is that they tend to develop similar solution architectures which combine the prediction of multiple independent modules and optionally handle domain constraints on top of the modules. These solution architectures have been shown empirically to work well. It will be interesting to see if such studies converge in a definitive blueprint architecture for making multiple inferences in knowledge-intensive domains.

Chapter 7

CONCLUSION

Representation matching is a critical step in numerous data management applications. Manual matching is very expensive. Hence, it is important to develop techniques to automate the matching process. Given the rapid proliferation and the growing size of applications today, automatic techniques for representation matching become ever more important.

This dissertation has contributed to both understanding the matching problem and developing matching tools. In this chapter, we recap the key contributions of the dissertation and discuss directions for future research.

7.1 Key Contributions

This dissertation makes two major contributions. The first contribution is a framework that formally defines a variety of representation-matching problems and explains the workings of subsequently developed matching algorithms.

The framework introduces a small set of notions: (1) a domain representation that serves as the user's conceptualization of the domain, (2) a mapping function that relates concepts in the representations to be matched to those in the domain representation, (3) a similarity function that the user employs to relate the similarity of concepts in the domain representation, (4) an assumption that relates the innate semantic similarity of concepts with their syntactic similarity, and (5) operators that are defined over concepts in the domain representation and that can be used to combine concepts to form complex mapping expressions.

We show that most types of input and output of representation matching problems (including output notions such as semantic mapping) can be explained in terms of the above five notions. An important consequence of this result is that it suggests a methodology to obtain input information about a matching problem by systematically checking what is known about each of the five notions. The more input information we have about a matching problem, the higher matching accuracy we can obtain.

The second major contribution of the dissertation is a solution to semi-automatically create semantic mappings. The key innovations that we made in developing this solution are:

- We brought the necessity of exploiting multiple types of information to the forefront of representation matching. Then we proposed a *multistrategy learning* solution, which applies multiple modules – each exploiting well a single type of information to make matching predictions – and then combines the modules' predictions. Employing *multiple independent matching modules* is a key idea underlying our solution, for both 1-1 and complex matching cases. This idea yields a solution that is highly modular and easily customized to any particular domain.

- We developed the A* and relaxation-labeling frameworks that exploit a broad range of integrity constraints and domain heuristics. These frameworks are made possible by our deci-

sion to layer constraint exploitation on top of the matching modules. (An alternative would have been to incorporate constraint handling directly into the modules.) Again, this two-layer architecture is modular and easily adapted to new domains, as we demonstrated by adapting our solution to data integration (Chapter 3), data translation (Chapter 4), and ontology matching (Chapter 5).

- We showed that explicit notions of similarity play an important part in practical matching scenarios. We then demonstrated that our solution can handle a broad variety of such notions (Chapter 5). This result is significant because virtually all previous works have not considered the notion of similarity explicitly.

- Finally, we showed that our solution can also naturally handle complex matchings, the types of matching that are common in practice but have not been addressed by most previous works. The first main idea here was to find a set of candidate complex mappings, then reduce the problem to an 1-1 matching problem. The second idea was to employ multiple search modules to examine the space of complex mappings, to find mapping candidates. The final main idea was to use machine learning and statistical techniques to evaluate mapping candidates.

7.2 Future Directions

We have made significant inroads into understanding and developing solutions for representation matching, but substantial work remains toward the goal of achieving a comprehensive matching solution. In what follows we discuss several directions for future work.

7.2.1 Efficient User Interaction

Matching solutions must interact with the user in order to arrive at final correct mappings. (Even if a solution is perfect, the user still has to verify the mappings.) We consider efficient user interaction *the* most important open problem for representation matching. Any practical matching tool must handle this problem, and anecdotal evidence abounds on deployed matching tools quickly being abandoned for irritating users with too many questions. Our experience with matching large schemas (e.g., while experimenting with the GLUE system) confirms that even just verifying a large number of created mappings is already extremely tedious.

The building and operating of future data sharing systems will further exacerbate this problem. Presumably many such systems will operate over hundreds or thousands of data sources. Even if a near perfect matching solution is employed, the system builder still has to verify the tens of thousands or millions of mappings that the solution created. Just the verification of mappings at such scales is already bordering on practical impossibility. Hence, efficient user interaction is crucial. The key is to discover how to minimize user interaction – asking only for absolutely necessary feedback, but maximizing the impact of the feedback.

7.2.2 Performance Evaluation

We have reported matching performance in terms of the predictive matching accuracy. Predictive accuracy is an important performance measure because (a) the higher the accuracy, the more reduc-

tion in human labor a matching system can achieve, and (b) the measure facilitates comparison and development of matching techniques. The next important task is to actually *quantify* the reduction in human labor that a matching system achieves. This problem is related to the problem of efficient user interaction that we mentioned above. It is known to be difficult, due to widely varying assumptions on how a matching tool is used, and has just recently been investigated [MMGR02, DMR02].

7.2.3 Unified Matching Framework

A third challenge is to develop a unified framework for representation matching that combines in a principled, seamless, and efficient way all the relevant information (e.g., user feedback, mappings from a different application) and techniques (e.g., machine learning, heuristics). The work on the GLUE system (Chapter 5) suggests that mappings can be given well-founded definitions based on probabilistic interpretations, and that a unified mapping framework can be developed by leveraging probabilistic representation and reasoning methods such as Bayesian networks.

7.2.4 Mapping Maintenance

In dynamic and autonomous environments (e.g., the Internet) sources often undergo changes in their schemas and data. Hence, the operators of a data sharing system must constantly monitor the component sources to detect and deal with changes in their semantic mappings. Clearly, manual monitoring is very expensive and not scalable. It is important therefore to develop techniques to automate the monitoring and repairing of semantic mappings. Despite the importance of this problem, it has not been addressed in the literature (though the related problem of wrapper maintenance has received some attention [Kus00b]).

7.2.5 Matching Other Types of Entities

Besides representation elements, the problems of matching other types of entities such as objects and Web services are also becoming increasingly crucial. The problem of deciding if two different objects in two sources (e.g., two house listings or two car descriptions) refer to the same real-world entity has received much attention in the database and data mining communities. This problem typically arises when multiple databases are merged and duplicate records must be purged (hence, it is also commonly known as the *merge/purge* problem). In the data integration context, the problem arises when we merge answers from multiple sources and must purge duplicate answers. As data integration becomes pervasive, this problem will become increasingly important.

The problem of deciding if two Web services share similar behaviors (in essence, matching the behaviors of services) will also become crucial as Web services proliferate and the need to mediate among them increases. It will be an interesting direction to examine how the techniques that have been developed for representation matching can be transferred to solving these new types of matching problems.

BIBLIOGRAPHY

[Agr90] A. Agresti. *Categorical Data Analysis*. Wiley, New York, NY, 1990.

[AK97] N. Ashish and C. Knoblock. Wrapper generation for semi-structured internet sources. *SIGMOD Record*, 26(4):8–15, 1997.

[BC86] J. Biskup and B. Convent. A formal view integration method. In *Proceedings of the ACM Conf. on Management of Data (SIGMOD)*, 1986.

[BCVB01] S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. Semantic integration of heterogeneous information sources. *Data and Knowledge Engineering*, 36(3):215–249, 2001.

[Ber03] P. Bernstein. Applying model management to classical meta data problems. In *Proceedings of the Conf. on Innovative Database Research (CIDR)*, 2003.

[BG00] D. Brickley and R. Guha. Resource description framework schema specification 1.0, 2000.

[BHP00] P. Bernstein, A. Halevy, and R. Pottinger. A vision for management of complex models. *ACM SIGMOD Record*, 29(4):55–63, 200.

[BKD+01] J. Broekstra, M. Klein, S. Decker, D. Fensel, F. van Harmelen, and I. Horrocks. Enabling knowledge representation on the Web by extending RDF schema. In *Proceedings of the Tenth Int. World Wide Web Conference*, 2001.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 279, 2001.

[BLN86] C. Batini, M. Lenzerini, and SB. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Survey*, 18(4):323–364, 1986.

[BM01] J. Berlin and A. Motro. Autoplex: Automated discovery of content for virtual databases. In *Proceedings of the Conf. on Cooperative Information Systems (CoopIS)*, 2001.

[BM02] J. Berlin and A. Motro. Database schema matching using machine learning with feature selection. In *Proceedings of the Conf. on Advanced Information Systems Engineering (CAiSE)*, 2002.

[CA99] S. Castano and V. De Antonellis. A schema analysis and reconciliation tool environment. In *Proceedings of the Int. Database Engineering and Applications Symposium (IDEAS)*, 1999.

[CDI98] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM SIGMOD Conference*, 1998.

[CGL01] D. Calvanese, D. G. Giuseppe, and M. Lenzerini. Ontology of integration and integration of ontologies. In *Proceedings of the 2001 Description Logic Workshop (DL 2001)*, 2001.

[CH98] W. Cohen and H. Hirsh. Joins that generalize: Text classification using WHIRL. In *Proc. of the Fourth Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, 1998.

[Cha00] H. Chalupsky. Ontomorph: A translation system for symbolic knowledge. In *Principles of Knowledge Representation and Reasoning*, 2000.

[CHR97] C. Clifton, E. Housman, and A. Rosenthal. Experience with a combined approach to attribute-matching across heterogeneous databases. In *Proc. of the IFIP Working Conference on Data Semantics (DS-7)*, 1997.

[CRF00] Donald D. Chamberlin, Jonathan Robie, and Daniela Florescu. Quilt: An XML query language for heterogeneous data sources. In *WebDB (Informal Proceedings) 2000*, pages 53–62, 2000.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, NY, 1991.

[dam] www.daml.org.

[DDH01] A. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: A machine learning approach. In *Proceedings of the ACM SIGMOD Conference*, 2001.

[DDH03] A. Doan, P. Domingos, and A. Halevy. Learning to match the database schemas: A multistrategy approach. *Machine Learning*, 2003. Special Issue on Multistrategy Learning. To Appear.

[DFF+99] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu. A query language for XML. In *Proceedings of the International Word Wide Web Conference, Toronto, CA*, 1999.

[DH74] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1974.

[DJMS02] T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining database structure; or, how to build a data quality browser. In *Proceedings of the ACM Conf. on Management of Data (SIGMOD)*, 2002.

[DMDH02] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map ontologies on the Semantic Web. In *Proceedings of the World-Wide Web Conference (WWW-02)*, 2002.

[DMR02] H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proceedings of the 2nd Int. Workshop on Web Databases (German Informatics Society)*, 2002.

[DP97] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[DR96] S. Donoho and L. Rendell. Constructive induction using fragmentary knowledge. In *Proc. of the 13th Int. Conf. on Machine Learning*, pages 113–121, 1996.

[DR02] H. Do and E. Rahm. Coma: A system for flexible combination of schema matching approaches. In *Proceedings of the 28th Conf. on Very Large Databases (VLDB)*, 2002.

[EJX01] D. Embley, D. Jackman, and L. Xu. Multifaceted exploitation of metadata for attribute match discovery in information integration. In *Proceedings of the WIIW Workshop*, 2001.

[EP90] AK. Elmagarmid and C. Pu. Guest editors' introduction to the special issue on heterogeneous databases. *ACM Computing Survey*, 22(3):175–178, 1990.

[Fen01] D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, 2001.

[Fre98] Dayne Freitag. Machine learning for information extraction in informal domains. *Ph.D. Thesis*, 1998. Dept. of Computer Science, Carnegie Mellon University.

[FW97] M. Friedman and D. Weld. Efficiently executing information-gathering plans. In *Proc. of the Int. Joint Conf. of AI (IJCAI)*, 1997.

[GMPQ+97] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. *Journal of Intelligent Inf. Systems*, 8(2), 1997.

[HGMN+98] J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. Breunig, and V. Vassalos. Template-based wrappers in the TSIMMIS system (system demonstration). In *ACM Sigmod Record*, Tucson, Arizona, 1998.

[HH01] J. Heflin and J. Hendler. A portrait of the Semantic Web in action. *IEEE Intelligent Systems*, 16(2), 2001.

[HNR72] P. Hart, N. Nilsson, and B. Raphael. Correction to "a formal basis for the heuristic determination of minimum cost paths". *SIGART Newsletter*, 37:28–29, 1972.

[HZ83] R.A. Hummel and S.W. Zucker. On the foundations of relaxation labeling processes. *PAMI*, 5(3):267–287, May 1983.

[iee01] *IEEE Intelligent Systems*, 16(2), 2001.

[IFF+99] Z. Ives, D. Florescu, M. Friedman, A. Levy, and D. Weld. An adaptive query execution system for data integration. In *Proc. of SIGMOD*, 1999.

[ILM+00] Z. Ives, A. Levy, J. Madhavan, R. Pottinger, S. Saroiu, I. Tatarinov, S. Betzler, Q. Chen, E. Jaslikowska, J. Su, and W. Yeung. Self-organizing data sharing communities with sagres. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, page 582, 2000.

[KMA+98] C. Knoblock, S. Minton, J. Ambite, N. Ashish, P. Modi, I. Muslea, A. Philpot, and S. Tejada. Modeling web sources for information integration. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.

[KSL+99] G. Keim, N. Shazeer, M. Littman, S. Agarwal, C. Cheves, J. Fitzgerald, J. Grosland, F. Jiang, S. Pollard, and K. Weinmeister. PROVERB: The probabilistic cruciverbalist. In *Proc. of the 6th National Conf. on Artificial Intelligence (AAAI-99)*, pages 710–717, 1999.

[Kus00a] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1–2):15–68, 2000.

[Kus00b] N. Kushmerick. Wrapper verification. *World Wide Web Journal*, 3(2):79–94, 2000.

[LC94] W. Li and C. Clifton. Semantic integration in heterogeneous databases using neural networks. In *Proceedings of the Conf. on Very Large Databases (VLDB)*, 1994.

[LC00] W. Li and C. Clifton. SEMINT: A tool for identifying attribute correspondence in heterogeneous databases using neural networks. *Data and Knowledge Engineering*, 33:49–84, 2000.

[LCL00] W. Li, C. Clifton, and S. Liu. Database integration using neural network: implementation and experience. *Knowledge and Information Systems*, 2(1):73–96, 2000.

[LG01] M. Lacher and G. Groh. Facilitating the exchange of explixit knowledge through ontology mappings. In *Proceedings of the 14th Int. FLAIRS conference*, 2001.

[Lin98] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1998.

[LKG99] E. Lambrecht, S. Kambhampati, and S. Gnanaprakasam. Optimizing recursive information gathering plans. In *Proc. of the Int. Joint Conf. on AI (IJCAI)*, 1999.

[Llo83] S. Lloyd. An optimization approach to relaxation labeling algorithms. *Image and Vision Computing*, 1(2), 1983.

[LRO96] A. Y. Levy, A. Rajaraman, and J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proc. of VLDB*, 1996.

[MBR01] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2001.

[MFRW00] D. McGuinness, R. Fikes, J. Rice, and S. Wilder. The Chimaera ontology environment. In *Proceedings of the 17th National Conference on Artificial Intelligence*, 2000.

[MHDB02] J. Madhavan, A. Halevy, P. Domingos, and P. Bernstein. Representing and reasoning about mappings between domain models. In *Proceedings of the National AI Conference (AAAI-02)*, 2002.

[MHH00] R. Miller, L. Haas, and M. Hernandez. Schema mapping as query discovery. In *Proc. of VLDB*, 2000.

[MHTH01] P. Mork, A. Halevy, and P. Tarczy-Hornoch. A model of data integration system of biomedical data applied to online genetic databases. In *Proceedings of the Symposium of the American Medical Informatics Association*, 2001.

[MMGR02] S. Melnik, H. Molina-Garcia, and E. Rahm. Similarity flooding: a versatile graph matching algorithm. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2002.

[MN98] A. McCallum and K. Nigam. A comparison of event models for Naive Bayes text classification. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[MS99] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*, pages 575–608. The MIT Press, Cambridge, US, 1999.

[MS01] A. Maedche and S. Saab. Ontology learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2), 2001.

[MT94] R. Michalski and G. Tecuci, editors. *Machine Learning: A Multistrategy Approach*. Morgan Kaufmann, 1994.

[MWJ] P. Mitra, G. Wiederhold, and J. Jannink. Semi-automatic integration of knowledge sources. In *Proceedings of Fusion'99*.

[MZ98] T. Milo and S. Zohar. Using schema matching to simplify heterogeneous data translation. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 1998.

[NHT+02] F. Neumann, CT. Ho, X. Tian, L. Haas, and N. Meggido. Attribute classification using feature analysis. In *Proceedings of the Int. Conf. on Data Engineering (ICDE)*, 2002.

[NM00] N.F. Noy and M.A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2000.

[NM01] N.F. Noy and M.A. Musen. Anchor-PROMPT: Using non-local context for semantic Matching. In *Proceedings of the Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

[NM02] NF. Noy and MA. Musen. PromptDiff: A fixed-point algorithm for comparing ontology versions. In *Proceedings of the Nat. Conf. on Artificial Intelligence (AAAI)*, 2002.

[Ome01] B. Omelayenko. Learning of ontologies for the Web: the analysis of existent approaches. In *Proceedings of the International Workshop on Web Dynamics*, 2001.

[ont] http://ontobroker.semanticweb.org.

[Pad98] L. Padro. A hybrid environment for syntax-semantic tagging, 1998.

[PB02] R. Pottinger and P. Bernstein. Creating a mediated schema based on initial correspondences. *IEEE Data Engineering Bulletin*, 25(3), 2002.

[PE95] M. Perkowitz and O. Etzioni. Category translation: Learning to understand information on the Internet. In *Proc. of Int. Joint Conf. on AI (IJCAI)*, 1995.

[PR00] V. Punyakanok and D. Roth. The use of classifiers in sequential inference. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS-00)*, 2000.

[PS98] C. Parent and S. Spaccapietra. Issues and approaches of database integration. *Communications of the ACM*, 41(5):166–178, 1998.

[PSTU99] L. Palopoli, D. Sacca, G. Terracina, and D. Ursino. A unififed graph-based framework for deriving nominal interscheme properties, type conflicts, and object cluster similarities. In *Proceedings of the Conf. on Cooperative Information Systems (CoopIS)*, 1999.

[PSU98] L. Palopoli, D. Sacca, and D. Ursino. Semi-automatic, semantic discovery of properties from database schemes. In *Proc. of the Int. Database Engineering and Applications Symposium (IDEAS-98)*, pages 244–253, 1998.

[PTU00] L. Palopoli, G. Terracina, and D. Ursino. The system DIKE: towards the semi-automatic synthesis of cooperative information systems and data warehouses. In *Proceedings of the ADBIS-DASFAA Conf.*, 2000.

[PVH+02] L. Popa, Y. Velegrakis, M. Hernandez, R. J. Miller, and R. Fagin. Translating web data. In *Proceedings of the Int. Conf. on Very Large Databases (VLDB)*, 2002.

[RB01] E. Rahm and P.A. Bernstein. On matching schemas automatically. *VLDB Journal*, 10(4), 2001.

[RD00] E. Rahm and H. Do. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 2000.

[RHS01] I. Ryutaro, T. Hideaki, and H. Shinichi. Rule induction for concept hierarchy alignment. In *Proceedings of the 2nd Workshop on Ontology Learning at the 17th Int. Joint Conf. on AI (IJCAI)*, 2001.

[RMR00] A. Rosenthal, F. Manola, and S. Renner. Getting data to applications: Why we fail, and how we can do better. In *Proceedings of the AFCEA Federal Database Conference*, 2000.

[RRSM01] A. Rosenthal, S. Renner, L. Seligman, and F. Manola. Data integration needs an industrial revolution. In *Proceedings of the Workshop on Foundations of Data Integration*, 2001.

[RS01] A. Rosenthal and L. Seligman. Scalability issues in data integration. In *Proceedings of the AFCEA Federal Database Conference*, 2001.

[SL90] AP. Seth and JA. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Survey*, 22(3):183–236, 1990.

[SR01] L. Seligman and A. Rosenthal. The impact of xml in databases and data sharing. *IEEE Computer*, 2001.

[SRLS02] L. Seligman, A. Rosenthal, P. Lehner, and A. Smith. Data integration: Where does the time go? *IEEE Data Engineering Bulletin*, 2002.

[TD97] L. Todorovski and S. Dzeroski. Declarative bias in equation discovery. In *Proceedings of the Int. Conf. on Machine Learning (ICML)*, 1997.

[TW99] K. M. Ting and I. H. Witten. Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10:271–289, 1999.

[UDB] UDB: The unified database for human genome computing. http://bioinformatics.weizmann.ac.il/udb.

[vR79] van Rijsbergen. *Information Retrieval*. London:Butterworths, 1979. Second Edition.

[Wol92] D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[Wor] Wordnet: A lexical database for the English language. http://www.cogsci.princeton.edu/ wn.

[XML98] Extensible markup language (XML) 1.0. www.w3.org/TR/1998/REC-xml-19980210, 1998. W3C Recommendation.

[Xqu] XQuery: An XML query language. http://www.w3.org/TR/xquery.

[XSL99] XSL Transformations (XSLT), version 1.0. http://www.w3.org/TR/xslt, 13 August 1999. W3C Working Draft.

[YMHF01] L.L. Yan, R.J. Miller, L.M. Haas, and R. Fagin. Data driven understanding and refinement of schema mappings. In *Proceedings of the ACM SIGMOD*, 2001.

[YS00] J. Yi and N. Sundaresan. A classifier for semi-structured documents. In *Proc. of the 6th Int. Conf. on Knowledge Discovery and Data Mining (KDD-2000)*, 2000.