Multifaceted Exploitation of Metadata for Attribute Match Discovery in Information Integration

David W. Embley, David Jackman, Li Xu
Department of Computer Science
Brigham Young University
Provo, Utah 84602, U.S.A.
embley@cs.byu.edu, djackman@nextpage.com, lx@cs.byu.edu

Abstract

Automating semantic matching of attributes for the purpose of information integration is challenging, and the dynamics of the Web further exacerbate this problem. Believing that many facets of metadata can contribute to a resolution, we present a framework for multifaceted exploitation of metadata in which we gather information about potential matches from various facets of metadata and combine this information to generate and place confidence values on potential attribute matches. To make the framework apply in the highly dynamic Web environment, we base our process largely on machine learning. Experiments we have conducted are encouraging, showing that when the combination of facets converges as expected, the results are highly reliable.

1 Introduction

In this paper, we focus on the long-standing and challenging problem of attribute matching [1] for the purpose of information integration. To address this problem, researchers have used a variety of techniques including the use of data values [2, 3], data-dictionary information [3], structural properties [4], ontologies [5], synonyms and other terminological relationships found in dictionaries and thesauri [6, 7, 8], and various combinations of these techniques [9, 10, 11]. These are the kinds of facets of metadata we wish to exploit, all of which may contribute to the resolution of attribute-matching issues. Although we probably have some idea about what metadata is most useful and in what combination and under what circumstances we should use this metadata, we probably do not know with certainty. Thus, rather than try to encode algorithms over the metadata ourselves, we largely use machine learning to develop the algorithms. This approach also has the advantage of being flexible in the presence of changing dynamics, which are so common on the Web.

As in [12], we assume that we wish to integrate data from multiple populated *source* schemes into a *target* scheme, where all schemes are described using the same conceptual model [13]. Ultimately, however, we can and do consider our sources to be Web repositories, which we reverse engineer into source schemes by the data-extraction processes we have defined for semistructured and unstructured Web pages [14], by the database reverse-engineering process we have defined, which works for Web tables and relational databases [15], and by the Web form data-extraction process we are developing [16]. Moreover, using standard representational transformations among conceptual-model schemes, we can transform the conceptual-model instance of any particular wrapper into a conceptual-model instance required by our technology, and thus we can make use of any developed wrapper technology (e.g. [17, 18, 19] and many more—see the bibliography in [14]). In addition to these assumptions for sources, we assume that target schemes are augmented with a variety of both application-independent and application-specific ontological information. For this paper the augmentations we discuss are WordNet [20, 21], which is application independent, sample data, which is application specific, and regular-expression recognizers, which are partly application specific and partly application independent.

Our contribution in this paper is the following framework which we propose as a way to discover which attributes in a source scheme S directly match with which attributes in a target scheme T.[1]

1. For each individual, independent facet, find potential attribute matches between the m attributes in S and the n attributes in T. Provide confidence measures between 0 (lowest confidence) and 1 (highest confidence) for each potential match. Section 2 explains how we generate matching rules over independent facets.

2. Using the confidence measures from (1), combine the measures for each potential match into a unified measure of match confidence. The result is an $m \times n$ matrix M of confidence measures. Section 3 explains how we combine confidence measures.

3. Iterate over M using a best-first match constrained by an injective assignment algorithm until all matches whose confidence measures exceed a threshold t are settled. Section 3 also explains how we settle attribute matches.

[1] In future work we intend to expand this framework to indirect matches in which target object and relationship sets match with virtual source object and relationship sets formed by queries over source model instances as set forth in [12], but we focus here only on direct attribute matches.

We illustrate our framework using car advertisements, which are plentiful on the Web, appearing in a variety of unstructured and structured forms. In Section 4 we report on the results obtained from this application, and in Section 5 we make concluding remarks.

2 Individual Facet Matching

In our framework we consider each individual facet separately. For each facet we obtain a vector of measures for the features of interest and then apply machine learning over this feature vector to generate a decision rule and a measure of confidence for each generated decision. We use C4.5 [22] as our decision-rule and confidence-measure generator.

So far we have investigated three facets: (1) terminological relationships (e.g. synonyms, word senses, and hypernym groupings), (2) data-value characteristics (e.g. average values, variances, string lengths), and (3) target-specific, regular-expression matches (i.e. whether expected strings appear in the data). We explain the details of these facets in the subsections below. We leave for future work the investigation of additional facets (e.g. data-dictionary descriptors, structural constraints, and scheme characteristics).

2.1 Terminological Relationships

One facet of metadata that usually gives humans a clue about which attributes to match are the meanings of the attribute names. To match attribute names, we need a dictionary or thesaurus. WordNet [20, 21] is a readily available lexical reference system that organizes English nouns, verbs, adjectives, and adverbs into synonym sets, each representing one underlying lexical concept. Other researchers have also suggested using WordNet to match attributes (e.g. [7, 23]), but have given few, if any, details.

Initially we investigated the possibility of using 27 available features of WordNet in an attempt to match an attribute A of a source scheme with an attribute B of a target scheme. The C4.5-generated decision tree, however, was not intuitive.[2] We therefore introduced some bias by selecting only those features we believed would contribute to a human's decision to declare a potential attribute match, namely (f0) same word (1 if $A = B$ and 0 otherwise), (f1) synonym (1 if "yes" and 0 if "no"), (f2) sum of the distances of A and B to a common hypernym ("is kind of") root (if A and B have no common hypernym root, the distance is defined as a maximum number in the algorithm), (f3) the number of different common hypernym roots of A and B, and (f4) the sum of the number of senses of A and B. For our training data we used 222 positive and 227 negative A-B pairs selected from attribute names found in database schemes readily available to us along with synonym names found in dictionaries. Figure 1(a) shows the resulting decision tree. Surprisingly, neither f0 (same word) nor f1 (synonym) became part of the decision rule. Feature f3 dominates—when WordNet cannot find a common hypernym root, the words are not related. After f3, f2 makes the most difference—if two words are closely related to the same hypernym root, they are a good potential match. (Note that f2 covers f0 and f1 because both identical words and direct synonyms have zero distance to a common root; this helps mitigate the surprise about f0 and f1.) Lastly, if the number of senses is too high (f4 $>$ 11), a pair of words tends to match almost randomly; thus the C4.5-generated rule rejects these pairs and accepts fewer senses only if pairs are reasonably close (f2 $<=$

[2] An advantage of decision-tree learners over other machine learning (such as neural nets) is that they generate results whose reasonableness can be validated by a human.

```
f3 <= 0: NO (222.0/26.0)
f3 > 0
|   f2 <= 2: YES (181.0/3.0)
|   f2 > 2
|   |   f4 <= 11
|   |   |   f2 <= 5: YES (15.0/5.0)
|   |   |   f2 > 5: NO (14.0/6.0)
|   |   f4 > 11: NO (17.0/2.0)
```

(a) WordNet Rule

	Car	*Year*	*Make*	*Model*	*Style*	*Payment*
Car	.98	.11	.11	.11	.12	.11
Year	.11	.98	.11	.11	.11	.11
Make	.11	.11	.98	.98	.98	.11
Model	.11	.11	.98	.98	.98	.11
Mileage	.11	.11	.11	.11	.11	.11
Phone	.43	.11	.11	.11	.43	.11
Price	.11	.11	.11	.11	.12	.98
Feature	.11	.11	.67	.12	.12	.11

(b) WordNet Matrix

Figure 1: Generated WordNet Rule and Confidence-Value Matrix

	Car	*Year*	*Make*	*Model*	*Style*	*Payment*
Car	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
Year	*NA*	.98	0	0	0	0
Make	*NA*	0	.97	.83	0	0
Model	*NA*	0	1	1	0	0
Mileage	*NA*	0	0	0	0	.97
Phone	*NA*	0	0	0	0	0
Price	*NA*	0	0	0	0	.14
Feature	*NA*	0	.05	.92	0	0

(a) Value Characteristics

	Car	*Year*	*Make*	*Model*	*Style*	*Payment*
Car	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
Year	*NA*	1	0	.04	0	.49
Make	*NA*	0	1	0	0	0
Model	*NA*	0	0	.87	.13	.01
Mileage	*NA*	0	0	0	0	0
Phone	*NA*	0	0	0	0	0
Price	*NA*	0	0	0	0	0
Feature	*NA*	0	0	.01	.99	0

(b) Expected Values

Figure 2: Confidence-Value Matrices

5) to a common root. The parenthetical numbers (x/y) following "YES" and "NO" for a decision-tree leaf L give the total number of training instances x classified for L and the number of incorrect training instances y classified for L.

Figure 1(b) shows a confidence-value matrix generated by the decision rule in Figure 1(a) for a sample application. The attributes on the top are source attributes taken from a Web table (www.swapaleas.com, November 2000).[3] The attributes on the left are target attributes taken from our standard car-ads data-extraction ontology (www.deg.byu.edu). For a "YES" leaf L, C4.5 computes confidence factors by the formula $(x - y)/x$ where x is the total number of training instances classified for L and y is the number of incorrect training instances classified for L.[4] For a "NO" leaf, the confidence factor is $1 - ((x - y)/x)$, which converts "NO's" into "YES's" with inverted confidence values. Observe that the confidence is high for the matches {Car, Car}, {Year, Year}, {Make, Make}, and {Model, Model}, as it should be. The confidence, however, is also high for {Make, Model}, {Make, Style}, and {Model, Style}, which are synonyms in some contexts, although not in car ads. Also, the confidence of {Price, Payment} is high, but "Price" is the selling price of a car, which should not match "Payment," the monthly payment of the lease. As we shall see, other facets are needed to sort out these differences.

[3] When attribute names were abbreviations, we expanded them so that WordNet could recognize them. We also selected nouns from phrase names. In future work, we intend to automate abbreviation expansion using dictionaries and noun selection using simple natural-language-processing techniques.

[4] We set the C4.5 parameter for rule-instance classification to 10 so that leaves with too few classifications would not have unsuitably high confidence factors.

2.2 Data-Value Characteristics

Another facet of metadata that usually gives humans a clue about which attributes to match is whether two sets of data, in some sense, have similar value characteristics. Previous work in [2] shows that this facet can successfully help match attributes by considering such characteristics as means and variances of numerical data and string-lengths and alphabetic/non-alphabetic ratios of alphanumeric data. We used the same features as in [2], but generated a C4.5 decision rule rather than a neural-net decision rule.

We trained the C4.5 decision-rule generator for our car-ads application using data from twenty-nine different car-ad Web sites scattered throughout the US. We generated two decisions trees, one for numeric data and one for alphanumeric data. Lacking space, we do not give the generated decision trees, which are similar in form to the decision tree in Figure 1(a) except that the alphanumeric decision tree is much larger. We do, however, give in Figure 2(a) the confidence-value matrix for our sample car-ads test case. In Figure 2(a) the "Car" attribute is a nonlexical attribute whose values are OID's, making them inapplicable for value analysis. Observe that years, makes, and models, which should match all have high confidence values. Observe, however, that the makes, models, and features all tend to look alike according to the value characteristics measured and that mileages and payments also look alike. These need to be further sorted out using other facets. Interestingly, prices and payments do not have similar value characteristics; this is because their means are vastly different.

2.3 Expected Data Values

Yet another facet of metadata that usually gives humans a clue about which attributes to match is the presence of expected data values. As explained in [12], we can associate with each attribute A in the target scheme a regular expression that matches values expected to appear for a source attribute B that potentially matches A. Furthermore, we can apply developed wrapper technology to extract a set of data values for the source attribute B. Then, using techniques described in [14], we can extract data values for source attributes and categorize them with respect to the attributes in the target, and thus match source and target attributes.

Instead of using C4.5 to generate a decision rule for expected data values, we directly generated confidence factors as follows. We applied the regular expression for each target attribute A against the set of values for each source attribute B and found the percentage of B values that matched (or included at least some match). Then, for each A-B pair, we simply let this percentage value be the confidence value. Figure 2(b) shows the matrix for our sample car-ads test case. Observe that years, makes, and models consistently include expected values, as expected. Further, makes, models, and styles do not get mixed up when we consider specific expected values—"Ford" is a make, not a model or a style; "Cavalier" is a model, not a make or a style; and "Sedan" is a style, not a make or a model. Interestingly, features and styles match—this is because features include styles in our car-ads ontology.

3 Combining Facets and Settling Matches

Although we would like to study more sophisticated combinations in the future, including the possibility of using machine learning to provide an appropriate decision rule, we cur-

	Car	*Year*	*Make*	*Model*	*Style*	*Payment*
Car	.98	.11	.11	.11	.12	.11
Year	.11	.99	.04	.05	.04	.20
Make	.11	.04	.99	.60	.33	.04
Model	.11	.04	.66	.95	.37	.04
Mileage	.11	.04	.04	.04	.04	.36
Phone	.43	.04	.04	.04	.14	.04
Price	.11	.04	.04	.04	.04	.38
Feature	.11	.04	.24	.35	.37	.04

(a) Combined Matrix

	Car	*Year*	*Make*	*Model*	*Style*	*Payment*
Car	1	0	0	0	0	0
Year	0	1	0	0	0	0
Make	0	0	1	0	0	0
Model	0	0	0	1	0	0
Mileage	0	0	0	0	.04	.36
Phone	0	0	0	0	.14	.04
Price	0	0	0	0	.04	.38
Feature	0	0	0	0	.37	.04

(b) Final Matrix

Figure 3: Combined Matrix and Final Confidence-Value Matrix with Settled Matches

```
Input: a matrix M of confidence values, and a threshold T.
Output: a set of matching attribute pairs.

While there is an unsettled confidence value in M greater than T
    Find the largest unsettled confidence value V in M;
    Settle V by setting it to 1;
    Mark V as being settled;
    For each unsettled confidence value W in the rows and columns of V
        Settle W by setting it to 0;
        Mark W as being settled;
Output the settled attribute pairs whose value is 1;
```

Figure 4: Attribute-Match Settling Algorithm

rently use a simple average over the confidence values for each attribute pair. Figure 3(a) shows the resulting combined matrix for our sample car-ads application.

We settle matching pairs by the algorithm in Figure 4, which is greedy (selects the highest confidence value first) and is an injective assignment algorithm (allows at most one match for any row or column). When we run this algorithm on the matrix in Figure 3(a) with a threshold value of 0.50, which we selected as an initial best guess based on the combined matrix, we obtain the final matrix in Figure 3(b). Observe that even though "Make-Model" pairs have values exceeding the threshold, the injective assignment constraint eliminates these matches because they are precluded by the "Make-Make" and "Model-Model" matches. Thus, the final matching pairs are {Car, Car}, {Year, Year}, {Make, Make}, and {Model, Model}, as they should be.

4 Experimental Results

In addition to our sample application presented here, we applied our method to six other car-ads tables found on the Web and obtained similar results. Over all test cases, the process matched 100% (32 of 32) of the direct matches. There were 2 false matches among a potential of 376 false matches—in one table "Feature" matched "Color," and in another "Feature" matched "Body Type." In our car-ads ontology, both colors and body types are special kinds of features, and thus the match was not entirely wrong—just not exact. In future work we need to verify these results across different applications with more complex schemes, but the results of these initial tests are indeed encouraging.

For comparison, we ran each individual facet matrix alone through the settling algorithm. In these tests, the settling process found only 90% of the direct matches (30 of 32 for WordNet, 21 of 25 applicable matches for value characteristics, and 23 of 25 applicable matches for expected values). The settling process also found 18 false matches (4 for WordNet, 8 for value characteristics, and 6 for expected values). These results suggest that the multifaceted approach proposed here is likely to be better than any single-faceted approach.

5 Concluding Remarks

We presented a framework for discovering direct matches between sets of source and target attributes. In the framework multiple facets each individually contribute in a combined way to produce a final set of matches. The results are encouraging and show that the multifaceted approach to exploiting metadata for attribute matching has promise.

References

[1] J. Larson, S. Navathe, and R. Elmasri. A theory of attribute equivalence in databases with application to schema integration. *IEEE Transactions on Software Engineering*, 15(4), 1989.

[2] W.-S. Li and C. Clifton. Semantic integration in heterogeneous databases using neural networks. In *Proceedings of the 20th Very Large Data Base Conference*, Santiago, Chile, 1994.

[3] E.H.C. Chua, R.H.L.Chiang, and E-P. Lim. Instance-based attribute identification in database integration. In *Proceedings of the 8th Workshop on Information Technologies and Systems (WITS'98)*, Helsinki, Finland, December 1998.

[4] M. Garcia-Solaco, F. Slator, and M. Castellanos. A structure based schema integration methodology. In *Proceedings of the 11th International Conference on Data Engineering (ICDE'95)*, pages 505–512, Taipei, Taiwan, 1995.

[5] J. Fowler, B. Perry, M. Nodine, and B. Bargmeyer. Agent-based semantic interoperability in InfoSleuth. *SIGMOD Record*, 28(1):60–67, March 1999.

[6] S. Hayne and S. Ram. Multi-user view integration system (MUVIS): An expert system for view integration. In *Proceedings of the 6th International Conference on Data Engineering*, pages 402–409, February 1990.

[7] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, March 1999.

[8] S. Castano and V. De Antonellis. Semantic dictionary design for database interoperability. In *Proceedings of 1997 IEEE International Conference on Data Engineering (ICDE'97)*, pages 43–54, Birmingham, United Kingdom, April 1997.

[9] V. Kashyap and A. Sheth. Semantic and schematic similarities between database objects: A context-based approach. *The VLDB Journal*, 5:276–304, 1996.

[10] V. Kashyap and A. Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In M. Papazoglou and G. Schlageter, editors, *Cooperative Information Systems: Current Trends and Directions*, pages 139–178, 1998.

[11] S. Castano, V. De Antonellis, M.G. Fugini, and B Pernici. Conceptual schema analysis: Techniques and applications. *ACM Transactions on Database Systems*, 23(3):286–333, September 1998.

[12] J. Biskup and D.W. Embley. Extracting information from heterogeneous information sources using ontologically specified target views. *Information Systems*. (to appear).

[13] D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[14] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.

[15] D.W. Embley and M. Xu. Relational database reverse engineering: A model-centric, transformational, interactive approach formalized in model theory. In *DEXA'97 Workshop Proceedings*, pages 372–377, Toulouse, France, September 1997. IEEE Computer Society Press.

[16] S.H. Yau. Automating the extraction of data behind web forms. Technical report, Brigham Young University, Provo, Utah, 2001. http://www.deg.byu.edu.

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

[18] P.B. Golgher, A.H.F. Laender, A.S. da Silva, and Ribeiro-Neto. An example-based environment for wrapper generation. In S.W. Liddle, H.C. Mayr, and B. Thalheim, editors, *Proceedings of the 2nd International Conference on the World-Wide Web and Conceptual Modeling*, Lecture Notes in Computer Science, 1921, pages 152–164, Salt Lake City, Utah, October 2000.

[19] J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. Breunig, and V. Vassalos. Template-based wrappers in the TSIMMIS system. In *Proceedings of 1997 ACM SIGMOD International Conference on Management of Data*, pages 532–535, Tucson, Arizona, May 1997.

[20] G.A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, November 1995.

[21] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachussets, 1998.

[22] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[23] S. Castano and V.D. Antonellis. ARTEMIS: Analysis and reconciliation tool environment for multiple information sources. In *Proceedings of the Convegno Nazionale Sistemi di Basi di Dati Evolute (SEBD'99)*, pages 341–356, Como, Italy, June 1999.