A Holistic Paradigm for Large Scale Schema Matching[*]

Bin He, Kevin Chen-Chuan Chang
Computer Science Department
University of Illinois at Urbana-Champaign
binhe@uiuc.edu, kcchang@cs.uiuc.edu

ABSTRACT

Schema matching is a critical problem for integrating heterogeneous information sources. Traditionally, the problem of matching multiple schemas has essentially relied on finding pairwise-attribute correspondences in isolation. In contrast, we propose a new matching paradigm, *holistic schema matching*, to match many schemas at the same time and find all matchings at once. By handling a set of schemas together, we can explore their *context* information that reflects the semantic correspondences among attributes. Such information is not available when schemas are matched only in pairs. As the realizations of holistic schema matching, we develop two alternative approaches: *global evaluation* and *local evaluation*. Global evaluation exhaustively assesses all possible "models," where a *model* expresses all attribute matchings. In particular, we propose the MGS framework for such global evaluation, building upon the hypothesis of the existence of a hidden schema model that probabilistically generates the schemas we observed. On the other hand, local evaluation independently assesses every single matching to incrementally construct such a model. In particular, we develop the DCM framework for local evaluation, building upon the observation that co-occurrence patterns across schemas often reveal the complex relationships of attributes. We apply our approaches to match query interfaces on the deep Web. The result shows the effectiveness of both the MGS and DCM approaches, which together demonstrate the promise of holistic schema matching.

1. INTRODUCTION

Schema matching (i.e., discovering semantically corresponding attributes in different schemas) is fundamental for enabling query mediation and data exchange across information sources [1, 15]. This article proposes a new type of schema matching, *holistic schema matching* and presents two alternative methods we developed recently as its realizations. Traditionally, schema matching has been approached mainly by finding *pairwise-attribute correspondence*, to construct an integrated schema for two or some (small number of) n sources. We observe that there are often challenges (and certainly also opportunities) to deal with large numbers of sources. In such scenarios, the challenge of large scale can itself be an opportunity for new approaches – We can take a holistic view of all the input schemas and find all matchings at once.

Such scenarios arise, in particular, for integrating databases across the Internet, or the so-called "deep Web." Our recent study [4] in April 2004 estimated 450,000 online databases. With the virtually unlimited amount of information, the deep Web is clearly an important frontier for data integration. On this deep Web, numerous online databases provide data via their *query interfaces*, instead of static URL links. Each query interface accepts queries over its *query schemas* (e.g., author, title, subject, ... for *amazon.com*). Schema matching across Web interfaces is essential for mediating queries across deep Web sources.

However, as Section 2 will discuss, existing schema matching works mostly focus on small scale integration by finding pairwise-attribute correspondences between two sources. To tackle the challenge of large scale schema matching, as well as to take advantage of its new opportunity, we propose a new paradigm, *holistic schema matching*, to match many schemas at the same time and find all matchings at once, as Figure 1 shows. In particular, holistic schema matching takes a set of schemas as input and outputs a semantic *model*, which contains all the matchings among the input schemas (e.g., a model of book schemas may contain author = writer = name, subject = category, ...). Such a holistic view enables us to explore the *context* information beyond two schemas (e.g., similar attributes across multiple schemas; co-occurrence patterns among attributes), which is not available when schemas are matched only in pairs.

Compared with traditional approaches, the holistic approach leverages the large scale to make schema matching more solvable– in particular, it enables effective exploration of the context information. Such context information will be abundant as more sources are exploited. Intuitively, we are building upon the "peer context" among schemas. Being context-based, the holistic matching will benefit from the scale: the accuracy will "scale" with the number of sources.

[*]This material is based upon work partially supported by NSF Grants IIS-0133199 and IIS-0313260. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

For instance, our specific MGS and DCM approaches are both statistical methods, which will thus benefit from more "observations."

With the holistic approach of schema matching, this article presents two alternative methods we developed recently as its realizations. Specifically, to realize holistic schema matching, we develop two different methods with respect to how a semantic model (as Figure 1 introduced) is discovered: *global evaluation* and *local evaluation*. Global evaluation assesses a model as a whole, while local evaluation incrementally constructs the model.

On one hand, global evaluation exhaustively evaluates all possible models and selects the best one among them. The best model contains the set of matchings with the highest overall confidence as the correct model.

In particular, we develop the MGS framework [7] for such global evaluation by hypothesizing the existence of a hidden generative model for each domain (e.g., Books, Movies) (Section 3). Under this hypothesis, a schema can be viewed as an instance generated from the model with some probabilistic behavior. Schema matching is thus transformed into the discovery of the hidden model, given a set of schema instances. Such hidden model discovery motivates us to develop the MGS framework, which discovers matchings with statistical hypothesis testing.

On the other hand, local evaluation independently assesses every single matching and then incrementally constructs a model (as a set of all matchings), instead of exhaustively enumerating all possible models. For instance, among all the potential matchings in Books, we may first select the most confident matching author = writer = name and consider it as part of the best model. Then we iteratively select the next most confident matching under this partial model result, toward eventually completing the best model.

In particular, we develop the DCM framework [9] for such local evaluation with the goal of extending the holistic idea to discover complex matchings among attributes in query interfaces (Section 4). In contrast to simple 1:1 matching, complex matching matches a set of m attributes to another set of n attributes, which is thus also called *m:n matching*. For instance, in Books, author is a synonym of the grouping of last name and first name, i.e., {author} = {first name, last name}; in Airfares, {passengers} = {adults, seniors, children, infants}. In the MGS framework, we built the generative modeling for simple 1:1 matchings; however, it is unclear, in global evaluation, how to extend such a model to accommodate complex matchings. To cope with complex matchings, we develop a local evaluation approach with the observation that co-occurrence patterns across schemas often reveal the complex relationships of attributes. We observe that *grouping attributes* (e.g., {first name, last name}) tend to be co-present in query interfaces and thus positively correlated. In contrast, *synonym attributes* are negatively correlated because they rarely co-occur. This insight motivates the DCM framework, which greedily discovers both simple 1:1 and complex matchings with a dual mining of positive and negative correlations.

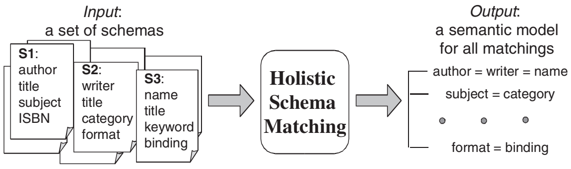

Figure 1: The holistic schema matching paradigm.

We compare global evaluation and local evaluation in Section 5. First, we qualitatively discuss their advantages and disadvantages. Second, we apply the MGS and DCM approaches to match Web query interfaces in the same domain (e.g., Books and Movies) and compare their matching accuracy.

The rest of the article is organized as follows: Section 2 reviews the related work. Section 3 briefly presents the MGS framework and Section 4 the DCM framework. Section 5 qualitatively discusses global evaluation and local evaluation and compares their experimental results on matching real query interfaces. Section 6 discusses some open issues and concludes the paper.

2. RELATED WORK

Traditionally, schema matching relies on matchings between pairwise attributes before integrating multiple schemas. For instance, traditional binary or n-ary [12] schema integration methodologies (as [1] surveys) exploit pairwise-attribute correspondence assertions (mostly manually given) for merging [13] or mapping [17] two or some n sources. Further, recent works on automatic schema matching mostly focus on matchings between two schemas (e.g., [11, 10]). Therefore, the latest survey [14] abstracts schema matching as pairwise similarity mappings between two input sources. In contrast, we propose a new type of schema matching, holistic schema matching, to match many sources at the same time and find all the matchings at once. Our work was motivated by integrating the deep Web, where the challenge of large scale matching is pressing. Our approaches leverage such scale to enable statistical analysis.

The closest idea to the holistic schema matching is probably the recent REVERE proposal [6], which suggests to use a separately-built schema corpus as a "knowledge-base" for assisting matching of unseen sources. While sharing the same insights of statistical analysis over corpora, our approach differs in that it leverages input schemas themselves as the "corpus."

Some recent work on matching Web query interfaces [16] also exploits the holistic idea. In particular, reference [16] explores syntactically similar attributes across multiple interfaces and develops clustering-based matching approaches. Essentially, this work is another realization of local evaluation of holistic schema matching by assessing matchings as cluster similarities.

3. GLOBAL EVALUATION: MATCHING AS HIDDEN MODEL DISCOVERY

To realize the global evaluation scheme, which finds an overall best model, we hypothesize the existence of the hidden generative behavior of a model. This hidden-model hypothesis provides a principled statistical method, hypothesis testing [2], to evaluate the confidence of a model (as a statistical hypothesis), given a set of schemas as observations. We thus abstract the schema matching problem as hidden model discovery and develop the MGS framework [7] to realize global evaluation.

In particular, our hidden-model hypothesis is based on two observations in our survey of the deep Web [4]: First, we observe *proliferating sources*: As the Web scales, many data sources exist to provide structured information in the same domains. Second, we also observe *converging vocabularies*: The aggregate schema vocabulary of sources in the same domain tends to converge at a relatively small size. That is, as sources proliferate, their vocabularies will tend to stabilize, which indicates that homogeneous sources (in the same domain) share some "concerted" vocabulary of attributes.

These observations lead us to hypothesize the existence of a hidden schema model that probabilistically generates, from a finite vocabulary, the schemas we observed. Intuitively, such a model gives the "structure" of the vocabulary to constrain how instances can be generated. The hypothesis sheds new light on a different way for coping with schema matching: If a hidden model does exist, its *discovery* would reveal the vocabulary structure. Such model-level unification of all attributes in the same domain will subsume their pairwise correspondence (as used in traditional schema matching). We thus propose the hidden model discovery approach as global evaluation for holistic schema matching.

More specifically, to realize such hidden model discovery, we develop a general abstract framework, MGS (for hypothesis *m*odeling, *g*eneration, and *s*election), with three steps: (1) *Hypothesis modeling*: We first specify a parameterized structure of the hypothetical hidden generative models. In particular, such models should capture the *target questions* of schema matching that we want to address– e.g., the model in Figure 2 focuses on simple 1:1 matchings. (2) *Hypothesis generation*: We then generate all "consistent" models that are likely to instantiate the observed schema instances. (3) *Hypothesis selection*: Finally, we select models of sufficient statistical consistency with the instances. Such an underlying model is likely the one that "generates" the input schemas, and thus its structure will answer our target integration questions.

To validate the idea of hidden model discovery by MGS, we performed an initial study: We specialize the MGS framework for finding simple 1:1 matchings for a sample of 55 book sources. We design a simple model to capture 1:1 matchings. Given the set of schema instances, MGS generates hypothetic models, under which it is "possible" (with non-zero probabilities) to "observe" these instances. Finally, we adopt χ^2 hypothesis testing [3] to select candidate models that are consistent with the instances at a sufficient significant level. The results are able to identify the hidden structure of 12 frequent attributes in the Books domain: {title (ti), author (au), ISBN (is), keyword (kw), publisher (pu), subject (su), last name (ln), format (fm), category (cg), price (pr), first name (fn), publication date (pd)}. This *vocabulary* collects all the attribute occurring at least in 10% of the 55 input sources.

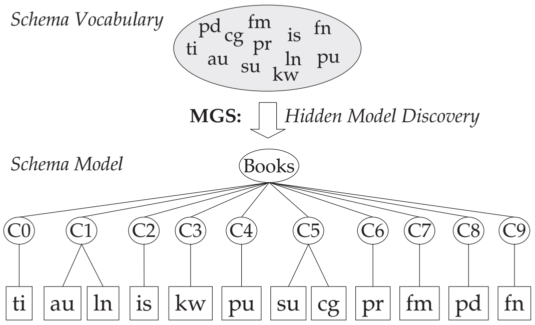

Figure 2: Case study of the Books domain.

Our initial study is very encouraging: Figure 2 shows the input unstructured vocabulary and the hidden model discovered by the MGS framework. The model has a structure that partitions the attributes into concepts ($C_0, \cdots, C_9$), each containing some synonym attributes– e.g., au and ln are synonyms in the same concept C_1. Note that, as a generative model, it can generate an instance schema by essentially picking one attribute from a concept. For simplicity, we omit the probabilistic parameters on the model, and only show the concept partition of attributes.

The discovered model correctly partitions the attributes into synonym attributes. The only error is that the model, while mapping ln to au in C_1, does not further group fn with ln for a complex matching to au – because the model was only designed to handle 1:1 matchings. In fact, the complete results (not shown here, see [7]) includes a second model, in which fn (but not ln) corresponds to au. Our study clearly indicates that, without any semantic annotations of the vocabulary, it is possible to achieve remarkable accuracy with a statistical framework. Please refer to [7] for more technical details and experimental evaluations.

4. LOCAL EVALUATION: MATCHING AS CORRELATION MINING

As discussed in Section 1, our realization of local evaluation deals with a more general type of matchings: complex matching. As local evaluation aims at "greedily" finding individual matchings (e.g., {author} = {first name, last name}), its realization relies on discovering some properties that indicate such matchings– We pursue a correlation mining approach by exploiting the *co-occurrence* patterns of attributes. Specifically, the holistic view provides the co-occurrence information of attributes across many schemas, which reveals the semantics of complex matchings. For instance, we may observe that last name and first name often co-occur in schemas, while they together rarely co-occur with author. More generally, we observe that *grouping attributes* (i.e., attributes in one group of a matching e.g., {last name, first name}) tend to be co-present and thus positively correlated across sources. In contrast, *synonym attributes* (i.e., attribute groups in a matching) are negatively correlated because they rarely co-occur in schemas.

These dual observations motivate us to develop a correlation mining abstraction of the schema matching problem. Specifically, we view a schema as a *transaction*, a conventional abstraction in association and correlation mining. In data mining, a transaction is a set of items; correspondingly, in schema matching, we consider a schema as a set of *attribute entities*. An attribute entity contains attribute name, type and domain (i.e., instance values). We develop a dual correlation mining framework, DCM, for mining complex matchings, consisting of three steps: mining positive correlations as groups, mining negative correlations as complex matchings and matching selection as model construction. In this section, we briefly summarize each step. Please refer to [9] for more details.

First, *group discovery*: We mine *positively correlated attributes* to form potential attribute groups. A potential group may not be eventually useful for matching; only the ones having synonym relationship (i.e., negative correlation) with other groups can survive. For instance, if all sources use last name, first name, and not author, then the potential group {last name, first name} is not useful because there is no matching (to author) needed.

Second, *matching discovery*: Given the potential groups (including singleton ones), we mine *negatively correlated attribute groups* to form potential complex matchings. A potential matching may not be considered as correct due to coincidental correlations. Specifically, as a statistical approach, correlation mining can discover true semantic matchings and, as expected, also false ones due to the existence of coincidental correlations. For instance, in Books domain, the mining result may have both {author} = {first name, last name}, denoted by M_1 and {subject} = {first name, last name}, denoted by M_2. We can see M_1 is correct, while M_2 is not. The reason for having the false matching M_2 is that in the collected schema data, it happens that subject does not often co-occur with first name and last name.

Third, *matching selection for model construction*: We develop an iterative selection algorithm to incrementally construct the model by choosing the most confident matching in each iteration. Specifically, the existence of false matchings may cause matching conflicts. For instance, M_1 and M_2 above conflict in that if one of them is correct, the other one will not. If both of them are correct, we should be able to also find the matching M_3: {author} = {subject} by the transitivity of synonym relationship. Since our mining algorithm does not find M_3, M_1 and M_2 cannot co-exist in the same model and thus they conflict. Based on this observation, we develop an iterative selection strategy to construct the model: In each iteration, we select the most confident matching as part of the best model and remove the conflicting matchings. By this iterative selection process, we incrementally construct a model with a set of consistent complex matchings.

Intuitively, between conflicting matchings, we want to select the more negatively correlated one because it indicates higher confidence to be synonyms. For example, our experiment shows that, as M_2 is coincidental, it is indeed that M_1 is more negatively correlated than M_2, and thus we select M_1 and remove M_2. With larger data size, semantically correct matching is more possible to be the winner. The reason is that, with larger size of sampling, the correct matchings are still negatively correlated while the false ones will remain coincidental and not as strong.

5. COMPARISONS

To better understand the characteristics of the MGS framework for global evaluation and the DCM framework for local evaluation, we compare these two approaches in both qualitative and experimental aspects.

5.1 Qualitative Analysis

Global evaluation is a more systematic and principled way to evaluate models since it exhaustively evaluates all possible models with a sound statistical basis. In particular, in the MGS framework, the statistical hypothesis testing can report matchings with respect to a given theoretical *significance level*. Also, the discovered model can naturally be employed as a unified schema to mediate queries to specific sources. However, global evaluation can be expensive. The exploration of all the possible models can be generally exponential, as shown in [7]. Further, modeling can be a difficult task, depending on specific target semantics to be discovered. In particular, it is unclear how to extend the modeling in [7] to accommodate complex matchings, which the DCM framework copes with (Section 4).

Local evaluation adopts a greedy strategy to incrementally construct a potentially suboptimal model. The greedy selection is not as systematic as the exhaustive enumeration in global evaluation. Also, as the core of correlation mining, we need to choose an appropriate correlation measure for our matching scenario. Since correlation measure is often empirically designed based on heuristics, the mining result may lack principled justification for its confidence. However, it does have some advantages. First, the computation of local evaluation is very efficient, since instead of exhaustively exploring all models, we select one matching at a time as part of the best model. Second, it is easier to accommodate complex matchings in local evaluation since it does not require formal statistical modeling.

domain	*the MGS framework*	*the DCM framework*
Books	{author} = {last name} (P) {author} = {first name} (P) {subject} = {category} (Y)	{author} = {last name, first name} (Y) {publisher} = {last name} (N) {subject} = {category} (Y)
Movies	{artist} = {actor} = {star} (Y) {genre} = {category} (Y)	{artist} = {actor} (P) {genre} = {category} (Y) {rating} = {keyword} (N) {price} = {format} (N)
MusicRecords	{title} = {album} (Y) {artist} = {band} (Y) {genre} = {soundtrack} (N) {keyword} = {catalog} (N)	{title} = {album} (Y) {artist} = {band} (Y) {genre} = {label} (N)
Automobiles	{style} = {type} = {category} (Y) {state} = {mileage} (N) {zip code} = {color} (N)	{style} = {type} = {category} (Y) {state} = {mileage} (N)

Figure 3: Experimental results of the two approaches on the BAMM dataset.

Domain	*Final Output After Matching Selection*	*Correct?*
Airfares	{destination (string)} = {to (string)} = {arrival city (string)}	Y
	{departure date (datetime)} = {depart (datetime)}	Y
	{passenger (integer)} = {adult (integer), child (integer), infant (integer)}	P
	{from (string), to (string)} = {departure city (string), arrival city (string)}	Y
	{from (string)} = {depart (string)}	Y
	{return date (datetime)} = {return (datetime)}	Y
Hotels	{check in (date), check out (date)} = {check in date (date), check out date (date)}	Y
	{check in (date)} = {check in date (date)}	Y
	{check out (date)} = {check out date (date)}	Y
	{type (string)} = {country (string)}	N
	{guest (integer)} = {adult (integer), child (integer), night (integer)}	P

Figure 4: Part of the experimental result of the DCM framework on the TEL-8 dataset.

5.2 Experimental Analysis

We apply the MGS and DCM frameworks in our motivating application: matching query interfaces on the deep Web, which is a special type of schema matching. Specifically, we choose two datasets, the BAMM dataset and the TEL-8 dataset, of the UIUC Web Integration Repository [5] as the testbed of our work. The BAMM dataset contains manually extracted attribute names over 211 sources in 4 domains (with around 50 sources per domain). The TEL-8 dataset contains raw Web pages over 447 deep Web sources in 8 popular domains. Each domain has about 20-70 sources.

Before matching, we perform simple a pre-processing step on the schemas by merging syntactically similar attributes (e.g., "title of book" is merged to "title"). In particular, we conduct a manual syntactic merging for the BAMM dataset and further fully automate this process for the TEL-8 databset by exploiting the syntactic similarity of both attribute names and instance values [9]. This pre-processing action serves for two purposes: First, it shows that syntactic merging cannot discover all the matchings, especially the "semantically difficult" ones. For instance, in Movies domain, star and actor are synonyms, but they bear essentially no syntactic similarity in names. Also, both of them are only associated with input boxes and thus have no instance values. As our experiment shows, many popular matchings are indeed "semantically difficult." Second, syntactic merging, by increasing the frequency of merged attributes, can enhance the accuracy of holistic matching approaches. The reason is that as statistical methods, these approaches rely on "sufficient observations" of attribute occurrences and thus they are likely to perform more favorably for frequent attributes.

In the experiments, we run the MGS and DCM frameworks on the BAMM dataset, to compare their ability in discovering simple 1:1 matchings. Also, to show the discovery of complex matchings, we test the DCM framework on the TEL-8 dataset. To illustrate the effectiveness of the holistic approaches, in this article, we only list and count the "semantically difficult" matchings discovered by the holistic algorithms, and not the "semantically simple" ones by the syntactic merging.

Results on the BAMM Dataset: We report the experimental results of the two approaches on the BAMM dataset, as Figure 3 shows. In particular, Figure 3 lists the discovered matchings for each of the four domains: Books, Movies, MusicRecords and Automobiles. A matching followed by "Y" means a correct matching, "P" a partially correct one and "N" an incorrect one. We can see that, in each domain, some correct matchings can be discovered by both approaches (e.g., {subject} = {category} in Books domain). Also, one approach may discover matchings that are not found by another. For instance, the DCM framework finds the complex matching {author} = {last name, first name} in Books domain, while the MGS framework currently only supports simple 1:1 matching and thus outputs two partially correct ones {author} = {last name} and {author} = {first name}. On the other hand, the MGS framework finds the fully correct matching {artist} = {actor} = {star} in MusicRecords domain, while the DCM framework only finds a partially correct one {artist} = {actor}. Finally, as statistical approaches, they may output some incorrect matchings due to the accidental bias of the data. From the results, we can see that, as two different matching evaluation approaches, global evaluation and local evaluation may not subsume each other, which indicates that combining their results may help improve the matching accuracy.

Results on the TEL-8 Dataset: In the BAMM dataset, only one complex matching is observed (i.e., {author} = {last name, first name}). However, in other domains such as Airfares, Hotels, CarRentals, more complex matchings can be found. To show the ability that the DCM framework can really discover complex matchings, we execute it on the 8 domains in the TEL-8 dataset, which contains more complex matchings. Because of the space limitation, we only show the discovered matchings in domains Airfares and Hotels, as Figure 4 shows. (The complete result can be found in [8].) The results show that the DCM framework can find complex matchings in many domains. For instance, in Airfares domain, we find 5 fully correct matchings, e.g., {from (string), to (string)} = {departure city (string), arrival city (string)}. Also, {passenger (integer)} = {adult (integer), child (integer), infant (integer)} is partially correct because it misses senior (integer). Note that since we incorporate type recognition in [9], the attribute names are followed by their data types in the matchings.

In summary, we can see that both approaches are effective and in some cases complementary in discovering "semantically difficult" matchings in Web query interfaces, which shows the promise of the holistic way of schema matching.

6. CONCLUDING DISCUSSION

In our study for holistic schema matching, we also observed some open issues that warrant further research. First, we plan to perform more thorough and systematic comparison for the two approaches. In this article, we compare them on the BAMM dataset. In the future, we plan to investigate a more systematic comparison. In particular, since the BAMM dataset only covers four domains with 50 sources in each domain, which may not be sufficient for thoroughly comparing the two approaches, a larger dataset with more domains and sources can be considered as the testbed. In addition to accuracy, it is also interesting to compare the two approaches on various other aspects, such as robustness to data noises (e.g., how accurate is the matching result if query interfaces, as input, are not perfectly extracted).

Second, given the respective pros and cons of global and local evaluations, we wonder if a hybrid of the two approaches will achieve the strength of both without the weakness of either. In particular, our goal is to design a hybrid approach with systematic modeling, rich expressiveness, and efficient execution. For instance, we can use the result of local evaluation to prune the search space of global evaluation. Or, we can use global evaluation to help the model construction of local evaluation. Specifically, in each iteration of greedy selection, instead of independently evaluating each potential matching, we can evaluate the confidence of incorporating each potential matching into the current partial model.

In summary, for large scale integration, our experience indicates a high promise for moving the traditional pairwise-attribute correspondence toward a new holistic schema matching approach. This approach is well suited for the new frontier of massive networked databases, such as the deep Web. In particular, we have developed the MGS and DCM frameworks as the realizations of global and local evaluations respectively for holistic schema matching.

7. REFERENCES

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

[2] P. J. Bickel and K. A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Prentice Hall, 2001.

[3] H. D. Brunk. *An Introduction to Mathematical Statistics*. New York, Blaisdell Pub. Co., 1965.

[4] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang. Structured databases on the web: Observations and implications. *SIGMOD Record*, 33(3), September 2004.

[5] K. C.-C. Chang, B. He, C. Li, and Z. Zhang. The UIUC web integration repository. Computer Science Department, University of Illinois at Urbana-Champaign. http://metaquerier.cs.uiuc.edu/repository, 2003.

[6] A. Halevy, O. Etzioni, A. Doan, Z. Ives, J. Madhavan, L. McDowell, and I. Tatarinov. Crossing the structure chasm. *Conf. on Innovative Database Research*, 2003.

[7] B. He and K. C.-C. Chang. Statistical schema matching across web query interfaces. In *SIGMOD Conference*, 2003.

[8] B. He, K. C.-C. Chang, and J. Han. Automatic complex schema matching across web query interfaces: A correlation mining approach. Technical Report UIUCDCS-R-2003-2388, Dept. of Computer Science, UIUC, Dec. 2003.

[9] B. He, K. C.-C. Chang, and J. Han. Discovering complex matchings across web query interfaces: A correlation mining approach. In *SIGKDD Conference*, 2004.

[10] Y. Lee, A. Doan, R. Dhamankar, A. Halevy, and P. Domingos. imap: Discovering complex mappings between database schemas. In *SIGMOD Conference*, 2004.

[11] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *VLDB Conference*, 2001.

[12] S. Navathe and S. Gadgil. A methodology for view integration in logical data base design. In *VLDB*, 1982.

[13] R. Pottinger and P. A. Bernstein. Merging models based on given correspondences. In *VLDB Conference*, 2003.

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

[15] L. Seligman, A. Rosenthal, P. Lehner, and A. Smith. Data integration: Where does the time go? *Bulletin of the Tech. Committee on Data Engr.*, 25(3), 2002.

[16] W. Wu, C. T. Yu, A. Doan, and W. Meng. An interactive clustering-based approach to integrating source query interfaces on the deep web. In *SIGMOD Conference*, 2004.

[17] L. L. Yan, R. J. Miller, L. M. Haas, and R. Fagin. Data-driven understanding and refinement of schema mappings. In *SIGMOD Conference*, 2001.