Data Fusion – Resolving Data Conflicts for Integration

Tutorial proposal, intended length 1.5 hours

Xin (Luna) Dong
AT&T Labs Inc.
Florham Park, NJ, USA
lunadong@research.att.com

Felix Naumann
Hasso Plattner Institute (HPI)
Potsdam, Germany
naumann@hpi.uni-potsdam.de

1. MOTIVATION

The amount of information produced in the world increases by 30% every year and this rate will only go up. With advanced network technology, more and more sources are available either over the Internet or in enterprise intranets. Modern data management applications, such as setting up Web portals, managing enterprise data, managing community data, and sharing scientific data, often require integrating available data sources and providing a uniform interface for users to access data from different sources; such requirements have been driving fruitful research on data integration over the last two decades [13, 15].

Data integration systems face two folds of challenges. First, data from disparate sources are often heterogeneous. Heterogeneity can exist at the schema level, where different data sources often describe the same domain using different schemas; it can also exist at the instance level, where different sources can represent the same real-world entity in different ways. There has been rich body of work on resolving heterogeneity in data, including, at the schema level, schema mapping and matching [17], model management [1], answering queries using views [14], data exchange [10], and at the instance level, record linkage (a.k.a., entity resolution, object matching, reference linkage, etc.) [9, 18], string similarity comparison [6], etc.

Second, different sources can provide conflicting data. Conflicts can arise because of incomplete data, erroneous data, and out-of-date data. Returning incorrect data in a query result can be misleading and even harmful: one may contact a person by an out-of-date phone number, visit a clinic at a wrong address, carry wrong knowledge of the real world, and even make poor business decisions. It is thus critical for data integration systems to resolve conflicts from various sources and identify true values from false ones. This problem becomes especially prominent with the ease of publishing and spreading false information on the Web and has recently received increasing attention.

This tutorial focuses on *data fusion*, which addresses the second challenge by fusing records on the same real-world entity into a single record and resolving possible conflicts from different data sources. Data fusion plays an important role in data integration systems: it detects and removes dirty data and increases correctness of the integrated data.

Objectives and Coverage. The main objective of the proposed tutorial is to gather models, techniques, and systems of the wide but yet unconsolidated field of data fusion and present them in a concise and consolidated manner. In the 1.5-hour tutorial we will provide an overview of the causes and challenges of data fusion. We will cover a wide set of both simple and advanced techniques to resolve data conflicts in different types of settings and systems. Finally, we provide a classification of existing information management systems with respect to their ability to perform data fusion.

Intended audience. Data fusion touches many aspects of the very basics of data integration. Thus, we expect the tutorial to appeal to a large portion of the VLDB community:

- *Researchers* in the fields of data integration, data cleansing, data consolidation, data extraction, data mining, and Web information management.
- *Practitioners* developing and distributing products in the data integration, data cleansing, ETL & data warehousing, and master data management areas.

We expect that attendees will take home from this seminar (i) an understanding of the causes and challenges of conflicting data along with different application scenarios, (ii) knowledge about concrete methods to resolve data conflicts both within relational DBMS and through dedicated applications, and (iii) an overview of existing tools and systems to perform data fusion.

Assumed background. Apart from a basic understanding of database technology and data integration, there are no prerequisites for this tutorial.

The proposed tutorial is based on a recent survey on data fusion [4] and various techniques proposed for truth discovery (including, but not limited to, [2, 7, 8, 19, 21]). We acknowledge the great contributions of authors of relevant papers.

2. TUTORIAL OUTLINE

Our tutorial starts from overviewing the importance of data fusion in data integration and possible reasons for data conflicts. We then present a classification of existing data fusion techniques and introduce relational operations for conflict resolution. After that, we describe several advanced

VLDB '09, August 24-28, 2009, Lyon, France

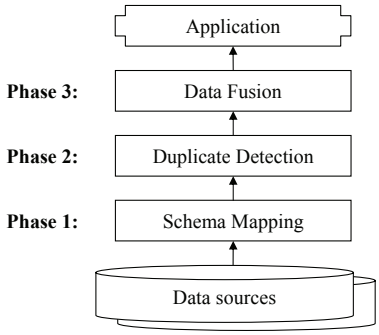

Figure 1: Three tasks in data integration (from [16]).

techniques for finding the best (true) values in presence of data conflicts. We end our tutorial with surveying data fusion techniques in existing data integration systems and suggesting future research directions.

2.1 Overview

Data integration has three broad goals: increasing the *completeness, conciseness,* and *correctness* of data that is available to users and applications. *Completeness* measures the amount of data, in terms of both the number of tuples and the number of attributes. *Conciseness* measures the uniqueness of object representations in the integrated data, in terms of both the number of unique objects and the number of unique attributes of the objects. Finally, *correctness* measures correctness of data; that is, whether the data conform to the real world.

Whereas high completeness can be obtained by adding more data sources to the system, achieving the other two goals is non-trivial. To meet these requirements, a data integration system needs to perform three levels of tasks (Fig. 1):

1. *Schema mapping:* First, a data integration system needs to resolve heterogeneity at the schema level by establishing semantic mappings between contents of disparate data sources.

2. *Duplicate detection:* Second, a data integration system needs to resolve heterogeneity at the instance level by detecting records that refer to the same real-world entity.

3. *Data fusion:* Third, a data integration system needs to combine records that refer to the same real-world entity by fusing them into a single representation and resolving possible conflicts from different data sources.

Among these three tasks, schema mapping and record linkage aim at removing redundancy and increasing conciseness of the data. Data fusion, which is the focus of this tutorial, aims at resolving conflicts from data and increasing correctness of data.

We distinguish two kinds of data conflicts: *uncertainty* and *contradiction*. *Uncertainty* is a conflict between a non-null value and one or more null values that are all used to describe the same property of a real-world entity. Uncertainty is caused by missing information, such as null values in a source or a completely missing attribute in a source. *Contradiction* is a conflict between two or more different non-null values that are all used to describe the same property of the same entity. Contradiction is caused by different sources providing different values for the same attribute of a real-world entity.

Table 1: Motivating example: five data sources provide information on the affiliations of five researchers. Only S_1 provides complete and correct information.

	S_1	S_2	S_3	S_4	S_5
Stonebraker	MIT	Berkeley	MIT	MIT	`null`
Dewitt	MSR	MSR	UWisc	UWisc	`null`
Bernstein	MSR	MSR	MSR	MSR	`null`
Carey	UCI	AT&T	BEA	BEA	BEA
Halevy	Google	Google	UW	UW	UW

EXAMPLE 2.1. *Consider the five data sources in Table 1. There exists uncertainty on the affiliations of* Stonebraker, Dewitt *and* Bernstein *because of the null values provided by source S_5, and contradiction on the affiliations of* Stonebraker, Dewitt, Carey *and* Halevy. □

There are two key issues in data fusion. First, how to find the best values among conflicting values? Second, how to do so efficiently? We next survey existing working on solving these problems.

2.2 Conflict resolution and data merging

Data conflicts, in the form of uncertainties or contradictions, can be resolved in numerous ways. After introducing a broad classification we turn to the relational algebra, which already provides several possibilities. More elaborate data integration systems and their fusion capabilities are analyzed in Section 2.4.

Conflict resolution strategies. There are many different data integration and fusion systems, each with their own solution. Fig. 2 classifies existing strategies to approach data conflicts and Table 2 lists some of the strategies and their classification. In particular, *Conflict ignoring* strategies are not aware of conflicts, perform no resolution, and thus may produce inconsistent results. *Conflict avoiding* strategies are aware of conflicts but do not perform individual resolution for each conflict. Rather, a single decision is made, e.g., preference of a source, and applied to all conflicts. Finally, *conflict resolving* strategies provide the means for individual fusion decisions for each conflict.

Such decisions can be *instance-based*, i.e., they regard the actual conflicting data values, or they can be *metadata-based*, i.e., they choose values based on metadata, such as freshness of data or the reliability of a source. Finally, strategies can be classified by the result they are able to produce: *deciding strategies* choose a preferred value among the existing values, while *mediating strategies* can produce an entirely new value, such as the average of a set of conflicting numbers.

Relational operations. Both *join* and *union* (and their relatives) perform data fusion of sorts. Joining two tables enlarges the schema of the original individual relations and thus appends previously unknown values to tuples. Outer-join variants avoid the loss of tuples without join partner. *Full disjunction* combines two or more input relations by combining all matching tuples into a single result-tuple [11].

Table 2: Conflict resolution strategies (from [3]).

Strategy	Classification	Short Description
PASS IT ON	ignoring	escalates conflicts to user or application
CONSIDER ALL POSSIBILITIES	ignoring	creates all possible value combinations
TAKE THE INFORMATION	avoiding, instance based	prefers values over null-values
NO GOSSIPING	avoiding, instance based	returns only "consistent" tuples
TRUST YOUR FRIENDS	avoiding, metadata based	takes the value of a preferred source
CRY WITH THE WOLVES	resolution, instance based, deciding	takes the most often occurring value
ROLL THE DICE	resolution, instance based, deciding	takes a random value
MEET IN THE MIDDLE	resolution, instance based, mediating	takes an average value
KEEP UP TO DATE	resolution, metadata based, deciding	takes the most recent value

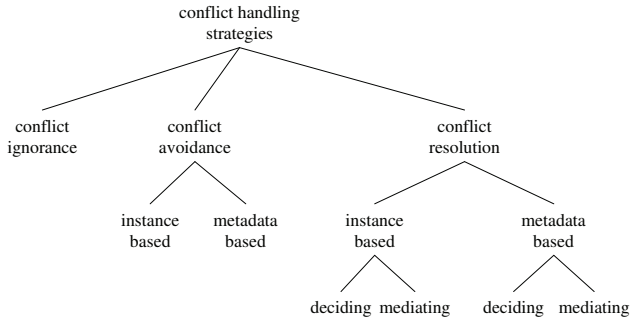

Figure 2: A classification of conflict resolution strategies (from [4]).

The union of two relations performs data fusion by fusing same tuples, i.e., pairs of tuples that have same values in all attributes. In the example of Tab. 1, a union of all 25 tuples would remove all exact duplicates and thus reduce the data set by 12 tuples. For instance, the fact that *Bernstein* works at *MSR* would be represented only once, increasing the readability of the result. A slight enhancement is given by the *minimum union* operation, which additionally removes subsumed tuples, i.e., tuples that agree with other tuples in all non-null values but have more null-values than the other. In the example, further 3 tuples, would be removed. For instance, the tuple *(Bernstein,null)* is subsumed by the tuple *(Bernstein,MSR)*. This definition is further extended to *complementing tuples*, i.e., tuples that have mutual uncertainties but no contradicting values [5]. For example, assume the tuples of the example had an additional attribute '*city*'. Tuples t_1 and t_2 in the following table are complementing tuples and would be fused to a more complete tuple:

t_1	*Bernstein*	*MSR*	`null`
t_2	*Bernstein*	`null`	*Redmond*
Fused result	*Bernstein*	*MSR*	*Redmond*

Besides removing uncertainties there have been relational approaches to remove contradictions. The *match-join* operator in a first step creates all possible tuples and in a second step reduces this number in a user-defined manner, for instance by selecting random tuples as a representative from a set of duplicates [20]. The *prioritized merge* operator [12] is similar but can give preferences to values of certain sources.

Finally, we discuss fusion through the SQL-based techniques of user-defined-functions, the coalesce function, and aggregation functions. All have the goal of resolving data conflicts by collecting possible values and producing a single, possibly new value for the fusion result.

2.3 Advanced techniques for conflict resolution

Obviously, none of the methods in Table 2 are perfect in resolving conflicts. They all fall short in some or all of the following three aspects. First, data sources are of different quality and we often trust data from more accurate sources, but accurate sources can make mistakes as well; thus, neither treating all sources as the same nor taking all data from accurate sources without verifying is appropriate. Second, the real world is dynamic and the *true* value often evolves over time (such as a person's affiliation and a business's contact phone number), but it is hard to distinguish incorrect values from out-of-date values; thus, taking the most common value may end up with an out-of-date value, whereas taking the most recent value may end up with a wrong value. Third, data sources can copy from each other and errors can be propagated quickly; thus, ignoring possible dependencies among sources can lead to biased decisions due to copied information.

We next describe several advanced techniques that consider accuracy of sources, freshness of sources, and dependencies between sources to solve the problems.

Considering accuracy of sources: Data sources are of different accuracy and some are more trustworthy. To illustrate, consider the first three sources in the motivating example. If we realize that S_1 is more accurate than the other two sources and give its values higher weights, we are able to make more precise decisions, such as correctly deciding that *Carey* is at *UCI* (there is a tie in voting between S_1, S_2, and S_3). It is proposed in [7, 19, 21] that we should consider accuracy of sources when deciding the true values. We describe their probabilistic models that iteratively compute accuracy of sources and decide the true values.

Considering freshness of sources: The world is often changing dynamically and a value, in addition to being true or false, can be in a subtle third case: *out-of-date*. Some sources, though appearing to provide wrong values, actually just have low freshness and provide stale data (S_3 in the motivating example falls in this category). It is proposed in [8] that we should consider *freshness* of sources and treat incorrect values and out-of-date values differently in truth discovery and we describe their probabilistic model accordingly.

Considering dependence between sources: In many

Table 3: Data fusion capabilities, possible strategies, and fusion specification in existing data integration systems (from [4]).

System	Fusion possible	Fusion strategy	Fusion specification
Multibase	resolution	Trust your friends, Meet in the middle	manually, in query
Hermes	resolution	Keep up to date, Trust your friends, ...	manually, in mediator
Fusionplex	resolution	Keep up to date	manually, in query
HumMer	resolution	Keep up to date, Trust your friends, Meet in the middle, ...	manually, in query
Ajax	resolution	various	manually, in workflow definition
TSIMMIS	avoidance	Trust your friends	manually, rules in mediator
SIMS/Ariadne	avoidance	Trust your friends	automatically
Infomix	avoidance	No Gossiping	automatically
Hippo	avoidance	No Gossiping	automatically
ConQuer	avoidance	No Gossiping	automatically
Rainbow	avoidance	No Gossiping	automatically
Pegasus	ignorance	Pass it on	manually
Nimble	ignorance	Pass it on	manually
Carnot	ignorance	Pass it on	automatically
InfoSleuth	unknown	Pass it on	unknown
Potter's Wheel	ignorance	Pass it on	manually, transformation

domains, especially on the Web, data sources may copy from each other for some of their data. In the motivating example, S_4 and S_5 copy all or part of the data from S_3. If we treat S_4 and S_5 the same as other sources, we will incorrectly decide that all data provided by S_3 are correct. It is proposed in [2, 7] that we should consider dependence between sources in truth discovery. We describe their algorithms that iteratively detect dependence between sources and discover the true values taking into consideration such dependence.

2.4 Data fusion in existing DI systems

This part of the tutorial examines relevant properties of both commercial and prototypical data fusion systems. The tutorial itself will not be held by rattling off long lists of properties and systems, but rather by highlighting certain relevant properties and special interesting features of these systems. The supplemental material can include the corresponding lists and tables found in [4]. An example is Tab. 3, which lists the fusion capabilities of different integration systems.

Among the analyzed research prototypes with some fusion capabilities are Multibase, Hermes, FusionPlex, HumMer, Ajax, TSIMMIS, SIMS, Ariadne, ConQuer, Infomix, HIPPO, and Rainbow (see [4] for references). Among the analyzed commercial data integration systems are several DBMS and ETL tools, such as IBM's Information Server or Microsoft's SQL Server Integration Services.

2.5 Open problems

We conclude the tutorial with a discussion of open problems and desiderata for data fusion systems. These include:

- *Complex fusion functions*: Often, the fusion decision is not based on the conflicting values themselves, but possibly on other data values of the affected tuples, such as a time stamp. In addition, fusion decisions on different attributes of the same tuples often need to coordinate, for instance in an effort to keep associations between first and last names and not to mix them from different tuples. Providing a language to express such fusion functions and developing algorithms for their efficient execution are open problems.

- *Incremental fusion*: Non-associative fusion functions, such as voting or average, are subject to incorrect results if new conflicting values appear. Techniques, such as retaining data lineage, maintaining simple metadata or statistics, need to be developed to facilitate incremental fusion.

- *Online fusion*: In some applications it is infeasible to fuse data from different sources in advance, either because it is impossible to obtain all data from some sources, or because the total amount of data from various sources is huge. In such cases we need to efficiently perform data fusion in an online fashion at the time of query answering.

- *Data lineage*: Database administrators and data owners are notoriously hesitant to merge data and thus lose the original values, in particular if the merged result is not the same as at least one of the original values. Retaining data lineage despite merging is similar to the problem of data lineage through aggregation operators. Effective and efficient management of data lineage in the context of fusion is yet to be examined.

- *Combining truth discovery and record linkage*: Although Fig. 1 positions data fusion as the last phase in data integration, the results of data fusion can often benefit other tasks. For example, correcting wrong values in some records can help link these records with records that represent the same entity. To obtain the best results in schema mapping, record linkage, and data fusion, we may need to combine them and perform them iteratively.

3. ABOUT THE PRESENTERS

Xin Luna Dong received a Bachelor's Degree in Computer Science from Nankai University in China in 1988, and a Master's Degree in Computer Science from Peking University in China in 2001. She obtained her Ph.D. in Compute Science and Engineering from University of Washington in 2007 and joined AT&T Labs–Research after graduation. Dr. Dong's research interests include databases, information retrieval

and machine learning, with an emphasis on data integration, data cleaning, probabilistic data management, schema matching, personal information management, web search, web-service discovery and composition, and Semantic Web. Dr. Dong has led development of the SEMEX personal information management system, which won the *best demo* award (one of the top 3 demos) in Sigmod'05.

Felix Naumann studied mathematics at the Technical University of Berlin and received his diploma in 1997. As a member of the graduate school "Distributed Information Systems" at Humboldt-University of Berlin, he finished his PhD thesis in 2000. His dissertation in the area of data quality received the dissertation prize of the German Society of Informatics (GI) for the best dissertation in Germany in 2000. In the following two years Felix Naumann worked at the IBM Almaden Research Center. From 2003-2006 he was an assistant professor at Humboldt-University of Berlin heading the Information Integration group. Since 2006 he is a full professor at the Hasso Plattner Institute, which is affiliated with the university of Potsdam. There he heads the information systems department. His experience in the area of data integration and data fusion is demonstrated by many publications in that area and numerous relevant industrial cooperations. Felix Naumann has served in the program committee of many international conferences and has served as a reviewer for many journals. He is the associate editor of the ACM Journal on Data and Information Quality and will be the general chair of the International Conference on Information Quality (ICIQ) in 2009.

4. REFERENCES

[1] P. A. Bernstein and S. Melnik. Model management 2.0: manipulating richer mappings. In *Proc. of SIGMOD*, pages 1–12, 2007.

[2] L. Berti-Equille, A. D. Sarma, X. Dong, A. Marian, and D. Srivastava. Sailing the information ocean with awareness of currents: Discovery and application of source dependence. In *CIDR*, 2009.

[3] J. Bleiholder and F. Naumann. Conflict handling strategies in an integrated information system. In *Proceedings of the International Workshop on Information Integration on the Web (IIWeb)*, Edinburgh, UK, 2006.

[4] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys*, 41(1):1–41, 2008.

[5] J. Bleiholder, S. Szott, M. Herschel, F. Kaufer, and F. Naumann. Algorithms for computing subsumption and complementation. 2009. Submitted.

[6] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proc. of IIWEB*, pages 73–78, 2003.

[7] X. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. Technical report, AT&T Labs–Research, Florham Park, NJ, 2009.

[8] X. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection from source update history. Technical report, AT&T Labs–Research, Florham Park, NJ, 2009.

[9] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(1):1–16, 2007.

[10] R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: Getting to the core. *ACM Transactions on Database Systems (TODS)*, 30(1):174–201, 2005.

[11] C. A. Galindo-Legaria. Outerjoins as disjunctions. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 348–358, Minneapolis, Minnesota, May 1994.

[12] S. Greco, L. Pontieri, and E. Zumpano. Integrating and managing conflicting data. In *Revised Papers from the 4th International Andrei Ershov Memorial Conference on Perspectives of System Informatics*, pages 349–362, 2001.

[13] L. M. Haas. Beauty and the beast: The theory and practice of information integration. In *Proc. of ICDT*, pages 28–43, 2007.

[14] A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.

[15] A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In *Proc. of VLDB*, pages 9–16, 2006.

[16] F. Naumann, A. Bilke, J. Bleiholder, and M. Weis. Data fusion in three steps: Resolving inconsistencies at schema-, tuple-, and value-level. *IEEE Data Engineering Bulletin*, 29(2):21–31, 2006.

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

[18] W. Winkler. Overview of record linkage and current research directions. Technical report, Statistical Research Division, U. S. Bureau of the Census, 2006.

[19] M. Wu and A. Marian. Corroborating answers from multiple web sources. In *Proc. of WebDB*, 2007.

[20] L. L. Yan and M. T. Özsu. Conflict tolerant queries in AURORA. In *Proceedings of the International Conference on Cooperative Information Systems (CoopIS)*, pages 279–290, 1999.

[21] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proc. of SIGKDD*, 2007.