Structures, Semantics and Statistics

Alon Y. Halevy
University of Washington, Seattle
alon@cs.washington.edu

Abstract

At a fundamental level, the key challenge in data integration is to reconcile the semantics of disparate data sets, each expressed with a different database structure. I argue that computing statistics over a large number of structures offers a powerful methodology for producing *semantic mappings*, the expressions that specify such reconciliation. In essence, the statistics offer hints about the semantics of the symbols in the structures, thereby enabling the detection of semantically similar concepts. The same methodology can be applied to several other data management tasks that involve search in a space of complex structures and in enabling the next-generation *on-the-fly* data integration systems.

Data Integration

Data integration is a pervasive challenge faced in data management applications that need to query across *multiple* data sources. Data integration is crucial in large enterprises that own a multitude of data sources, for progress in large-scale scientific projects, where data sets are being produced independently by multiple researchers, for better cooperation among government agencies, each with their own data sources, and in searching the *deep web*, the part of the web that is hidden behind web forms. The emergence of XML and web services as technologies for sharing data and for accessing remote data sources have further fueled the desire of organizations to share data. The many applications of data integration have led to a very fruitful line of research in the Database and Artificial Intelligence Communities, and recently to a budding industry, known as Enterprise Information Integration (EII) [1].

Structures and Semantics

There are many factors that make data integration a hard problem, not all of which are purely technical. Some of these include query processing across multiple autonomous systems, processing XML documents (and other semi-structured data) streaming from the network, managing data ownership and privacy across organizational boundaries, and in some cases, even capturing or locating the data needed for particular applications, or transforming it into machine processable form. However, the most notable and unique challenge in data integration is reconciling the semantic heterogeneity of the sources being integrated.

The fundamental reason that makes semantic heterogeneity so hard is that the data sets were developed independently, and therefore varying structures were used to represent the same or overlapping concepts. By structures, I mean both the choice of data model (relational, XML, object-oriented, ontology formalism) and the particular choices made in designing the schema (naming of relations, attributes or tags, choices of data types, decomposition, and nesting structure). The presence of a variety of structures is unavoidable both because humans think differently from one another and because the applications these data sets were designed for have different needs. Efforts to circumvent this problem by imposing standardized schemas have met limited success at best.

As a first step toward reconciling semantic heterogeneity, researchers developed languages for describing *semantic mappings*, expressions that relate the semantics of data expressed in different structures [6, 9, 15]. These languages typically relate different structures with a variety of query and constraint expressions. With these languages, researchers have developed algorithms for *query reformulation*, which translate a query posed over one schema into a set of queries over other schemas. More generally, recent research on *model management* investigates a general algebra for manipulating structures (called models) and mappings between them [2, 13]. The algebra includes operations such as merging and applying transformations on models, and for composing and inverting mappings.

**Proceedings of the 30th VLDB Conference,
Toronto, Canada, 2004**

Structures and Statistics

Given the languages for expressing semantic mappings between disparate structures, the bottleneck is to *create* and *maintain* these mappings. Writing these mappings is very tedious and error prone, and often very repetitive. In fact, in many integration projects, more than half of the resources are spent on these tasks. Clearly, completely automating the creation of semantic mappings is unlikely. Hence, the focus of research has been on reducing the human effort needed in the process (see [14] for a recent survey).

This is where statistics come into play. A powerful approach for discovering semantic mappings is based on analyzing a large number of structures and mappings in a particular domain. The intuition behind this approach is that statistics computed over large number of structures can be used to provide *hints* about the semantics of the symbols used in these structures. Therefore, these statistics can be leveraged to predict when two symbols, from disparate structures, are meant to represent the same domain concept.

In a sense, the goal of this approach is to mirror the success of statistical analysis of large corpora of texts in the field of Information Retrieval (IR) and of the recent significant advances made in the field of Natural Language Processing by analyzing large corpora of annotated sentences [12]. However, the analogy to these fields also highlights the unique challenges we face here.

In the IR context, text documents typically contain a significant amount of information and high level of redundancy. Hence, IR techniques can be effective by abstracting a document as a bag of words. By contrast, in our context, schema descriptions are very terse and the underlying semantics are very rich. Hence, the bag of words abstraction does not suffice.

As a consequence, to exploit a corpus of schemas and mappings, we need statistics that provide hints about deeper domain concepts and at a finer granularity. The following are a few examples:

- **Domain concepts and their representational variations:** As a first step, we can analyze a corpus to identify the main concepts in the domain. For example, in a corpus of book inventory schemas, we may identify the concept of book and warehouse and a cluster of price-related elements. Even more importantly, we will discover *variations* on how these concepts are represented. The variations may differ on naming of schema elements, grouping attributes into tables or the granularity of modeling a particular concept. Knowledge of these variations will be leveraged when we match two schemas in the domain.

- **Relationships between concepts:** Given a set of concepts, we can discover relationships between them, and the ways in which these relationships are manifested in the representation. For example, we can find that the Books table typically includes an ISBN column and a foreign key into an Availability table, but that ISBN never appears in a Warehouse table. These relationships are useful in order to prune candidate schema matches that appear less likely. They can also be used to build a system that provides advice in *designing* new schemas.

- **Domain constraints:** We can leverage a corpus to find integrity constraints on the domain and its representations. For example, we can observe that ISBN is a foreign key into multiple tables involving books, and hence possibly an identifier for books, or discover likely data types for certain fields (e.g., address, price). Constraints may have to do with *ordering* of attributes. For example, in a corpus of web forms about cars for sale, we may discover that the make attribute is always placed before the model and price attribute, but occurs after the new/used attribute.

  Typically, constraints we discover in this way are *soft constraints*, in the sense that they are sometimes violated, but can still be taken as rules of thumb about the domain. Therefore, they are extremely useful in resolving ambiguous situations, such as selecting among several candidate schema matches [3, 11].

It is important to note that in all of these examples there is a close interplay between properties of the underlying domain (e.g., books, warehouses and their properties) and of the representations of the domain (e.g., the particular relational structures in schemas). In fact, this interplay is the reason this technique is so powerful.

Several works have already applied this approach in various contexts [3, 5, 7, 8, 10]. Doan et al. [3] address the problem of matching schemas of data sources to a single mediated schema. [3] uses Machine Learning techniques to compute models of the elements in the mediated schema from a set of manually provided mappings. These models are then used to recognize the mediated schema elements in the schemas of unmapped data sources. He and Chang [7] generate a mediated schema for a domain based on analyzing a corpus of web forms in that domain. Madhavan et al. [11] leverage a corpus of schemas and mappings to match between two *unseen* schemas. In doing so, [11] learns from the corpus models for elements of the domain and constraints on the domain.

Another application of this paradigm is search for web services [4]: locating web services (or operations within them) that are relevant to a particular need. Simple keyword search does not suffice in this context because keywords (or parameter names) do not capture the underlying semantics of the web service. Dong et al.[4] show how to analyze a corpus of web services and cluster parameter names into semantically meaningful concepts. These concepts are used to predict when two web service operations have similar functionality.

Searching for web services is an instance of a general class of search problems, where the objects being searched have rich semantics, but the descriptions of these objects (e.g., schema definitions or WSDL descriptions) are terse and do not fully capture their semantics. Other examples of such search problems are in trying to locate web forms that are relevant to a particular information need, or locating relevant data sources within an enterprise. In all of these examples, simple keyword search does not suffice. Analyzing a corpus of such objects, and using the statistics to glean hints about the semantics of the objects offers a powerful supplement to keyword search. I now outline a major challenge for the field of data integration which will benefit significantly from this general approach.

A Data Integration Challenge

Despite the immense progress, building a data integration application is still a major undertaking that requires significant resources, upfront effort and technical expertise. As a result, data integration systems have two major drawbacks. First, evolving the system as the requirements in the organization change is hard. Second, many smaller-scale and more transient information integration tasks that we face on a daily basis are not supported.

Hence, a challenge to our community is to fundamentally change the cost-benefit equation associated with integrating data sources. Our goal should be to enable *on-the-fly* data integration, thereby facilitating the evolution of data integration applications and enabling individuals to easily integrate information for their personal, possibly transient, needs.

To achieve this goal, I believe a data integration environment should incorporate the following two principles. First, as data integration tasks are performed, the system should accumulate and analyze them, and then leverage prior tasks when facing a new task. Second, the data integration environment should be a natural extension of the user's *personal* information space, i.e., the information one stores on the desktop. In that way, a user can extend her personal data space with public data sources, and seamlessly integrate personal information (e.g., spreadsheets, contacts lists, personal databases) with organizational resources. Achieving these goals will substantially increase the perception of data management systems and their impact on our daily lives.

Acknowledgements

The ideas espoused in this paper have benefited from many discussions and hard work by my colleagues and students. I'd like to thank Phil Bernstein, Anhai Doan, Pedro Domingos, Luna Dong, Oren Etzioni, Zack Ives, Jayant Madhavan, Luke McDowell, Peter Mork, Rachel Pottinger, Dan Suciu, Peter Tarczy-Hornoch and Igor Tatarinov. The work has been supported by NSF ITR grant IIS-0205635, NSF CAREER Grant IIS-9985114 and a gift from Microsoft Research.

References

[1] Aberdeen Group. Enterprise information integration – the new way to leverage e-information. Aberdeen Group, Boston, Mass., 2003.

[2] P. A. Bernstein. Applying Model Management to Classical Meta Data Problems. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2003.

[3] A. Doan, P. Domingos, and A. Y. Halevy. Reconciling Schemas of Disparate Data Sources: A Machine Learning Approach. In *Proceedings of the ACM SIGMOD Conference*, 2001.

[4] X. L. Dong, A. Y. Halevy, J. Madhavan, E. Nemes, and J. Zhang. Similarity search for web services. In *Proc. of VLDB*, 2004.

[5] A. Halevy, O. Etzioni, A. Doan, Z. Ives, J. Madhavan, L. McDowell, and I. Tatarinov. Crossing the structure chasm. In *Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR)*, 2003.

[6] A. Y. Halevy. Answering Queries Using Views: A Survey. *VLDB Journal*, 10(4), 2001.

[7] B. He and K. C.-C. Chang. Statistical Schema Matching across Web Query Interfaces. In *Proceedings of the ACM SIGMOD Conference*, 2003.

[8] A. Hess and N. Kushmerick. Learning to Attach Semantic Metadata to Web Services. In *Proceedings of the International Semantic Web Conference*, 2003.

[9] M. Lenzerini. Data Integration: A Theoretical Perspective. In *In Proceedings of PODS*, 2002.

[10] J. Madhavan, P. Bernstein, K. Chen, A. Halevy, and P. Shenoy. Matching schemas by learning from others. In *Working notes of the IJCAI-03 workshop on Data Integration on the Web*, 2003.

[11] J. Madhavan, P. Bernstein, A. Doan, and A. Halevy. Corpus-based schema matching. Technical Report 2004-06-04, University of Washington, 2004.

[12] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[13] S. Melnik, E. Rahm, and P. Bernstein. Rondo: A programming platform for generic model management. In *Proc. of SIGMOD*, 2003.

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

[15] J. D. Ullman. Information Integration using Logical Views. In *Proceedings of the International Conference on Database Theory (ICDT)*, 1997.