19 The Future of Data Integration

"Prediction is very difficult, especially if it's about the future."
— **Niels Bohr**

Like the entire field of data management, data integration keeps evolving as technology shifts and new applications come to the forefront. Initially, data integration was focused on challenges that occur within enterprises. As the Web emerged, new problems were introduced to data integration, such as scaling to a much larger number of sources and handling sources with less structure. We continue to see the field evolving today as data are being produced faster than ever before and in new forms. Large scientific experiments rely on data management for progress. People create data fragments (often called breadcrumbs) whenever they interact with services on the Web. Sensors are collecting data from more sources (e.g., GPS, vehicle movement, vital signs) and in higher fidelity than ever before. Since data sources are created by multiple people and organizations, new sources of data end up yielding new challenges to data integration.

As we have seen throughout the book, data integration tends to be one of the drivers of new subfields of database management and a beneficiary of techniques developed in these fields. As a few examples, we have seen this in the development of XML, uncertain databases, provenance, keyword search on databases, and Web data management, each of which was developed for independent reasons but contributed in some way to data integration.

In the following paragraphs we touch upon some of the trends we see in data integration and areas that require additional attention.

19.1 Uncertainty, Provenance, and Cleaning

In some sense, the fact that data integration takes a set of heterogenous sources, transforms and combines then, and produces a uniform result can be both a plus and a minus. Sometimes knowing where the data have come from and how they were produced is critical. Perhaps a particular information extractor turns out to have some issues. Perhaps a source is especially authoritative.

As discussed in Chapter 14, data provenance relates source data items to results, while also recording the transformations applied. Given some information about the quality of some of the source (or result) values, plus the provenance, we would like to be able to reason about the the uncertainty we should have in the other data values.

Principles of Data Integration, DOI: 10.1016/B978-0-12-416044-6.00019-3

Existing work has looked at how to compute probabilistic answers given scores on the inputs, as discussed in Chapter 13. Some early work has also been done on learning from user feedback on the quality of the query results, as discussed in Chapter 16.

More generally, one would like to take a wide array of general data cleaning and update operations — deduplication, value correction, etc. — and use them, plus provenance, to revise our ranking of a source's quality (and possibly even to create general repair scripts). In general, a problem that is likely to see increasing attention is how to take usage, feedback, and data quality cues and use them to assess uncertainty and automate cleaning. A second key issue is how to develop formalisms, akin to the data models and query primitives we use over data, that can nicely capture the semantics of the uncertainty propagation operations we wish to perform.

19.2 Crowdsourcing and "Human Computing"

Inspired by the success of Amazon Mechanical Turk and Web 2.0 sites such as Wikipedia, many researchers have started investigating crowdsourcing as basic database operator. The key observation is that some conditions are hard for a computer to evaluate (e.g., whether an image contains a sunset, or whether an extraction from a Web page is correct), but can be done trivially by humans. Hence, if human evaluation can be built in to query processing (even if it takes longer), then new kinds of queries and data management services can be supported, such as finding images that satisfy a certain set of conditions and obtaining clean databases from Web pages.

A second kind of crowdsourcing requires the crowd members to be more proactive. For example, imagine building a database that contains the price of bottled water anywhere in the world, or finding which villages in Tanzania have access to clean water. Traditionally, creating such databases could take multiple years, and by the time they were completed they would be out of date. With appropriate crowdsourcing technology powered by mobile devices, such databases can be created within hours, and their effects can be transformative on people's lives.

Crowdsourcing has the potential of providing powerful solutions to traditionally hard data integration problems. For example, the quality of extractions from the Web can be verified by humans, and schemas and data can be matched by crowds. Work on this topic is in its infancy, but will undoubtedly be a focus in the upcoming years.

19.3 Building Large-Scale Structured Web Databases

In many domains there is a need to build large-scale structured Web databases by integrating data from many different sources. For example, Google Scholar integrates many bibliographic sources to build a citation database for millions of publications. As another

example, biologists often want to integrate sources such as PubMed, Grants, and ClinicalTrials to build biomedial databases. As yet another example, e-commerce companies often integrate product information from vendor feeds to build giant product taxonomies for searching and advertising purposes.

Building and maintaining such databases raise numerous challenges. Many of these challenges, such as wrapper construction and schema matching, have been discussed in this book. Many others are new, and we have only begun to understand them, as we build more and more databases. For example, we need to develop a methodology for building such databases. If we are to integrate PubMed, Grants, and ClinicalTrials, how should we proceed? Should we clean each source first (e.g., via data matching), then merge them, or should we do the reverse? There are well-understood methodologies for building relational databases. But no such methodology exists for building large-scale databases via integration.

As another example, when deploying such databases on the Web, we often create a "home page" for each integrated entity. However, as we gain more information, we often split or merge entities. How do we manage the IDs of such entities, and how do we provide navigational pages so that users can easily find the desired entities, even after we have moved them around? Wikipedia has developed a basic solution for these problems. But can it be used in other contexts? And are there better solutions? Other challenges include incrementally updating the database and recycling human feedback during the update process. Current research on building portals has begun to identify and address such challenges (see Chapter 15), but far more work is necessary.

19.4 Lightweight Integration

At the other end of the spectrum, many data integration tasks are transient. We often face a task where we need to integrate data from multiple sources to answer a question that will be asked only once or twice. However, the integration needs to be done quickly and by people who may not have much technical expertise. For example, consider a disaster response situation in which reports are coming from multiple data sources in the field, and the goal is to corroborate them and quickly share them with the affected public.

In general, there are many easy-to-explain integration tasks that are surprisingly difficult to perform using current tools and techniques. Beyond the major challenge of creating a data integration system that is usable by indviduals with a wide range of technical skills, lightweight integration introduces several other challenges. These include locating relevant data sources, assessing the quality of the sources, helping the user understand the semantics well enough so the sources can be integrated in a meaningful fashion, and supporting the process of integration.

One key to progress in this area is to focus on the task(s) the user is facing and make sure that each of the steps is easily supported. In addition, as we describe in Section 15.5,

we should support pay-as-you-go data integration: this should offer an immediate and clear return on the investment in reconciling, cleaning, and integrating the data. Ideally, machine learning and other techniques can be used to amplify the effects of human input, possibly through *semi-supervised learning*, where small amounts of human data classification, plus large amounts of additional raw ("unlabeled") data, are used to train the system.

19.5 Visualizing Integrated Data

Ultimately, users do not want to view rows of data but rather visualizations that highlight the important patterns in the data and offer flexible exploration. For example, a map or a timeline can quickly reveal a hot region of interest or a point in time that requires detailed investigation. While visualization is a challenge for the entire field of data management, it becomes even more important in data integration. For example, when data come from multiple sources we would like to immediately see discrepancies between the sources. In addition, a visualization can also aid in the process of integration itself by pinpointing a subset of the data that was not reconciled correctly. As people start searching through large collections of heterogeneous data, visualizations will be key to presenting the different search results and evaluating their relevance to the task at hand. Finally, visualizations also play a key role in conveying the certainty of the integrated data and its provenance.

19.6 Integrating Social Media

Social media includes Facebook updates, tweets, user-generated videos, and blogs. Such data are exploding, and there is much interest in integrating them. One integration scenario is to discover interesting events in the Twittersphere. Examples include a recent earthquake, the emerging negative reception of a newly released product, and a planned protest in a public square. Another integration scenario is to find all social media data (e.g., tweets and videos) that are related to a particular event. Yet another integration scenario is to summarize and visualize major trends in the blogosphere.

Clearly, integrating social media can bring tremendous benefits. However, it also raises many difficult challenges, due to the particular nature of the medium. First, social media data are often noisy, full of spam and low-quality data. Second, identifying quality data and influential users in social media is often quite difficult due to the transient nature of such data and users. Third, the data often lack context, making them hard to interpret and integrate. For example, it is difficult to interpet the tweet "mel just crashed his maserati" unless we know that Mel Gibson just had an accident. Finally, social media data often arrive as high-speed streams that require very fast processing. This raises not just "big data" but also "fast data" challenges. Work on integrating social media is just emerging, but will undoubtedly be a major focus of the field of data integration in the coming years.

19.7 Cluster- and Cloud-Based Parallel Processing and Caching

The ultimate vision of the data integration field is to be able to integrate large numbers of sources with large amounts of data — ultimately approaching the scale of the structured part of the Web.

We are currently far from this vision. Most query engines, schema matchers, storage systems, query optimizers, and so forth have been developed for operation on a single server or a small set of servers and quickly run into performance limitations. Likewise, many core algorithms are based on assumptions of limited scale (e.g., they are limited to main memory).

Ultimately, the ability to scale up to more sources, as well as to spend more processing power to make more accurate matches and more extensive searches, will almost certainly require us to redesign many algorithms to exploit the power of large clusters. Problems like schema matching, entity resolution, data cleaning, indexing, and so on will likely need to be tackled in a much more parallelizable and scalable way.