Introduction to the Special Issue on Semantic Integration

AnHai Doan\textsuperscript{1}, Natalya F. Noy\textsuperscript{2}, Alon Y. Halevy\textsuperscript{3}

\textsuperscript{1}University of Illinois, \textsuperscript{2}Stanford University, \textsuperscript{3}University of Washington
anhai@cs.uiuc.edu, noy@smi.stanford.edu, alon@cs.washington.edu

1 Introduction

Semantic heterogeneity is one of the key challenges in integrating and sharing data across disparate sources, data exchange and migration, data warehousing, model management, the Semantic Web and peer-to-peer databases. Semantic heterogeneity can arise at the schema level and at the data level. At the schema level, sources can differ in relations, attribute and tag names, data normalization, levels of detail, and the coverage of a particular domain. The problem of reconciling schema-level heterogeneity is often referred to as schema matching or schema mapping. At the data level, we find different representations of the same real-world entities (e.g., people, companies, publications, etc.). Reconciling data-level heterogeneity is referred to as data deduplication, record linkage, and entity/object matching. To exacerbate the heterogeneity challenges, schema elements of one source can be represented as data in another. This special issue presents a set of articles that describe recent work on semantic heterogeneity at the schema level.

Despite its pervasiveness and the substantial amount of work in this area, schema matching today remains a very difficult problem. In practice, schema matching is done manually with the help of graphical user interfaces. However, the process is known to be error-prone and labor intensive. It is common that 60-80\% of the resources in a data sharing project are spent on reconciling semantic heterogeneity.

This short article has two purposes. First, we provide some background on schema matching. Second, we briefly introduce the papers in the rest of the issue. The article is not meant to be a survey of schema matching. We refer the reader to several surveys written over the years [17, 16, 1, 8], as well as a companion special issue of the AI Magazine that contains a short survey of the field [7].

2 Problem Definition

The first step of reconciling schema-level semantic heterogeneity is to find correspondences (also called matches) between the two schemas. The matches can either be 1-1 matches between elements of the two schemas, or many-to-1 (or many-to-many). In Figure 1, for example, a 1-1 semantic match between the two relational databases on house listing and the semantic matches between them. Database \textit{S} consists of two tables \textit{HOUSES} and \textit{AGENTS}; database \textit{T} consists of table \textit{LISTINGS}.

After semantic matches have been found, we proceed to find the mapping expressions that enable translating data from one source to the other. Depending on the context, the expression can be written in languages such as SQL, LAV & GLAV [10, 13], XQuery or even simply in code. For more details on the phase of creating mappings, we refer the reader to [19].

3 Matching Techniques

There are several dimensions along which schema matching techniques differ: the similarity functions, the information taken into consideration in matching, and the architecture of the overall system.

3.1 Measuring Similarity

A plethora of techniques have been employed for determining similarity among two schemas. Many techniques employ hand-crafted rules or natural heuristics (see [17] for a survey). Examples of such heuristics include linguistic matching of names of schema elements, detecting overlap in the choice of data types and representation data values, considering patterns in relationships between elements, propagating similarity of related schema elements, using domain knowledge. Several recent approaches employ machine learning techniques. Approaches that leverage ideas from the fields of information theory and information retrieval have also been proposed [4, 12].

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{schema_matching.png}
\caption{The schemas of two relational databases on house listing, and the semantic matches between them. Database \textit{S} consists of two tables \textit{HOUSES} and \textit{AGENTS}; database \textit{T} consists of table \textit{LISTINGS}.}
\end{figure}
3.2 Information Used in Matching

In the earliest work, matching techniques exploit only schema-related evidence such as attribute names, their types, and integrity constraints. Subsequently, techniques were developed that exploit data-related evidence, such as the characteristics of value distributions of attributes (min, max, average, etc.), and the frequency of words in data instances.

More recently, several works have proposed using external evidence beyond the two schemas being matched. There are several examples of external evidence, such as past matches [6, 5, 2, 17], corpora of schemas and matches in a particular domain [14, 11, 18]. In fact, we can also leverage a large collection of contributors to supply mappings [15].

Finally, several works have leveraged detailed models or ontologies of a particular domain to enhance the matching system. Ontologies are written using representation languages developed in the knowledge representation community. They enable detailed and rich description of the properties that hold in a particular domain. We note that the AI community has considered the ontology matching problem for some time. In principle, this is a similar problem to schema matching, but the goal is to match much richer structures.

3.3 System Architecture

Most of the early schema matching systems exploit only certain types of information, and employ only certain similarity measurement techniques. A recent emerging consensus is that a robust matching system must exploit all available types of information to maximize matching accuracy. To this end, several recent works [6, 5, 9] have described a system architecture that employs multiple modules called matchers, each of which exploits well a certain type of information to predict matches. The system then combines the predictions of the matchers to arrive at a final prediction for matches. The combination can be manually specified [5] or automated to some extent using learning techniques [6]. In addition to being able to exploit multiple types of information, the multi-matcher architecture has the advantage of being very modular and can be easily customized to a new application domain.

4 Current Challenges

While there has been significant progress in developing schema matching systems, the current challenges have to do with deploying these systems in the real world. To that end, there are three main challenges that need to be addressed:

- Scalability with the size of the schema: most of the research has focused on small to medium sized schemas. Significant work is needed to extend these techniques to deal with large schemas. In fact, the basic paradigm of predicting a complete match between two given schemas may need to be reconsidered, and a more piece-meal approach may be preferable.

- User interaction: a schema matching system will never be completely autonomous, and will always need to interact with a user to create a correct mapping. When the schema is large, the system needs to focus the attention of the user to the parts of the schema that are relevant to their task, and to guide them in finding matches to that part of the schema. Finally, schema matching may be one component of a larger task that the user is performing (see, for example, scenarios using model management systems in [3]). Hence, a significant challenge we currently face is embedding the matching technology developed thus far into a useful context in which users can perform their tasks.

- Mapping maintenance: one of the reasons that schema matching is such an important problem is that schemas change frequently, and therefore mappings need to be maintained. Adapting schema matching techniques to facilitate maintenance is also a significant challenge.

5 Overview of the Issue

The papers in this special issue fall into four categories.

**Schema matching techniques:** The paper “Automatic Direct and Indirect Schema Mapping: Experiences and Lessons Learned” by Emlley et al. describes a system that uses a rich domain model in schema matching. The paper “A Holistic Paradigm for Large Scale Schema Matching” by He and Chang describes settings where one must match multiple schemas all at once. Here the knowledge gleaned from each matching pair can help match other pairs. As a result we can obtain better accuracy than just matching a pair in isolation.

**Current technical challenges:** The paper “Matching Large XML Schemas” by Rahm et al. focuses on the challenges of scaling schema matching to large schemas. Specifically, it describes matching very large and complex XML schemas in commercial enterprise settings. The paper “Kanata: Adaptation and Evolution in Data Sharing Systems” by Andritsos et al. addresses the problem of mapping maintenance at both the schema- and data levels. The paper also addresses
the problem of querying data that has become inconsistent as a result of changes.

**Practical considerations:** The paper “Industrial-Strength Schema Matching” by Bernstein et al. describes Protoplasm, an effort to build an industrial strength schema matching system. The system adopts and extends the multi-matcher architecture described earlier, but tries to build an environment in which schema matching can be done as part of other model management tasks. The paper “From Semantic Integration to Semantics Management: Case Studies and a Way Forward” by Rosenthal et al. argues that to make the greatest impact, we must go beyond after-the-fact semantic integration among existing systems, to actively guiding semantic choices in new ontologies and systems (e.g., what concepts should be used to describe existing data, or to define newly created data and systems). The paper “Multiplex, Fusionplex, and Autoplex, Three Generations of Information Integration” by Motro et al. describes a series of projects that consider semantic issues in building data integration systems. The paper addresses both finding semantic mappings as well as resolving data inconsistencies that arise when merging data from multiple sources.

**The role of ontologies:** Several recent trends indicate that ontologies will play a greater role in data management in the future. To that end, we decided to include two papers that offer a glimpse into the differences between ontologies and schemas, their role in schema matching and the challenges of ontology matching. The paper “Ontologies and Semantics for Seamless Connectivity” by Uschold and Gruninger discusses the often-mentioned simplification that ontologies are just “database schemas on steroids”. It explains what ontologies are and are not, and gives a brief overview of recent developments in the ontology community. The second paper, “Semantic Integration: a Survey of Ontology-Based Approaches” by Noy continues where the first paper ends, and gives a brief overview of the semantic integration tools in the ontology community.

**References**


