Advances in Ontology Matching

Avigdor Gal
Technion – Israel Institute of Technology
avigal@ie.technion.ac.il

Pavel Shvaiko
University of Trento, Povo, Trento, Italy
pavel@dit.unitn.it

Abstract

Matching of concepts describing the meaning of data in heterogeneous distributed information sources, such as database schemas and other metadata models, grouped here under the heading of an ontology, is one of the basic operations of semantic heterogeneity reconciliation. The aim of this chapter is to motivate the need for ontology matching, introduce the basics of ontology matching, and then discuss several promising themes in the area as reflected in recent research works. In particular, we focus on such themes as uncertainty in ontology matching, matching ensembles, and matcher self-tuning. Finally, we outline some important directions for future research.

1 Introduction

Matching of concepts describing the meaning of data in heterogeneous distributed information sources (*e.g.*, database schemas, XML DTDs, HTML form tags) is one of the basic operations of semantic heterogeneity reconciliation. Due to the cognitive complexity of this matching process [18], it has traditionally been performed by human experts, such as web designers, database analysts, and even lay users, depending on the context of the application [79, 47]. For obvious reasons, manual concept reconciliation in dynamic environments such as the web (with or without computer-aided tools) is inefficient to the point of being infeasible, and so cannot provide a general solution for semantic reconciliation. The move from manual to semi-automatic matching has therefore been justified in the literature using arguments of scalability, especially for matching between large schemas [45], and by the need to speed-up the matching process. Researchers also argue for moving to fully-automatic, that is, unsupervised, schema matching in settings where a human expert is absent from the decision process. In particular, such situations characterize numerous emerging applications, such as agent communication, semantic web service composition,

triggered by the vision of the semantic web and machine-understandable web resources [9, 82].

As integration of distributed information sources has been made more automated, the ambiguity in concept interpretation, also known as semantic heterogeneity, has become one of the main obstacles to this process. Heterogeneity is typically reduced in two steps: (*i*) matching of concepts to determine alignments and (*ii*) executing the alignment according to application needs (*e.g.*, schema integration, data integration, query answering). In this chapter, we focus only on the first, *i.e.*, the matching step, automation of which still requires much research. The second step has already found a certain level of support from a number of commercial tools, such as Altova MapForce[1] and BizTalk Schema Mapper.[2]

In the context of web applications and the advent of the semantic web, a new term, in addition to schema matching, has come into existence, namely *ontology matching*. Ontologies are considered to be semantically richer than schemas in general, and therefore, techniques for schema matching can be easily adopted to ontologies but not vice versa. Therefore, in this chapter, unless explicitly referenced, we consider schema matching to be a special case of ontology matching.

Research into schema and ontology matching has been going on for more than 25 years now (see surveys [5, 79, 69, 73, 81] and various online lists, *e.g.*, OntologyMatching[3], Ziegler[4], DigiCULT[5], and SWgr[6]) first as part of a broader effort of schema integration and then as a standalone research. Recently, ontology matching has been given a book account in [30]. This work provided a uniform view on the topic with the help of several classifications of the available methods, discussed these methods in detail, *etc.* The AI-complete nature of the problem dictates that semi-automatic and automatic algorithms for schema and ontology matching will be largely of heuristic nature. Over the years, a significant body of work was devoted to the identification of automatic matchers and construction of matching systems. Examples of state of the art matching systems include COMA [21], Cupid [55], OntoBuilder [35], Autoplex [8], Similarity Flooding [58], Clio [60, 43], Glue [22], S-Match [37, 39], OLA [31], Prompt [66] and QOM [27] to name just a few. The main objective of these is to provide an alignment, namely a set of correspondences between semantically related entities of the ontologies. It is also expected that the correspondences will be effective from the user point of view, yet computationally efficient or at least not disastrously expensive. Such research has evolved in different research communities, including artificial intelligence, semantic web, databases, information retrieval, information sciences, data semantics, and others. We have striven to absorb best matching experiences of these communities and report here in a

[1] `http://www.altova.com/products/mapforce/data_mapping.html`
[2] `http://msdn2.microsoft.com/en-us/library/ms943073.aspx`
[3] `http://www.ontologymatching.org/`
[4] `http://www.ifi.unizh.ch/~pziegler/IntegrationProjects.html`
[5] `http://www.digicult.info/pages/resources.php?t=10`
[6] `http://www.semanticweb.gr/`

uniform manner some of the most important advances.

The aim of this chapter is to motivate the need for ontology matching (Section 2), introduce the basics of ontology matching (Section 3), and then discuss several promising directions in the area as reflected in recent research works. In particular, we focus on the following themes: uncertainty in ontology matching (Section 4), matching ensembles (Section 5), and matcher self-tuning (Section 6). Finally, we conclude with a summary and outline some directions for future research (Section 7).

2 Applications

Matching ontologies is an important task in traditional applications, such as ontology integration, schema integration, and data warehouses. Typically, these applications are characterized by heterogeneous structural models that are analyzed and matched either manually or semi-automatically at design time. In such applications matching is a prerequisite of running the actual system.

A line of applications that can be characterized by their dynamics, *e.g.*, agents, peer-to-peer (P2P) systems, and web services, is emerging. Such applications, contrary to traditional ones, require (ultimately) a run time matching operation and often take advantage of more explicit conceptual models.

Below, we first discuss a motivating example and give intuition about the matching operation and its result. It is presented in the settings of the schema integration task. Then, we discuss data integration as yet another example of a traditional application. Finally, we overview a number of emergent applications, namely, P2P information systems, web service composition, and query answering on the deep web.

2.1 Motivating example

To motivate the matching problem, let us use two simple XML schemas ($O1$ and $O2$) that are shown in Figure 1 and exemplify one of the possible situations which arise, for example, when resolving a schema integration task [80].

Let us suppose an e-commerce company needs to finalize a corporate acquisition of another company. To complete the acquisition we have to integrate databases of the two companies. The documents of both companies are stored according to XML schemas $O1$ and $O2$, respectively. Numbers in boxes are the unique identifiers of the XML elements. A first step in integrating the schemas is to identify candidates to be merged or to have taxonomic relationships under an integrated schema. This step involves ontology matching. For example, the entities with labels Office_Products in $O1$ and in $O2$ are the candidates to be merged, while the entity with label Digital_Cameras in $O2$ should be subsumed by the entity with label Photo_and_Cameras in $O1$. Once correspondences between two schemas have been determined, the next step will generate query expressions that automatically translate data instances of these schemas under an integrated schema.

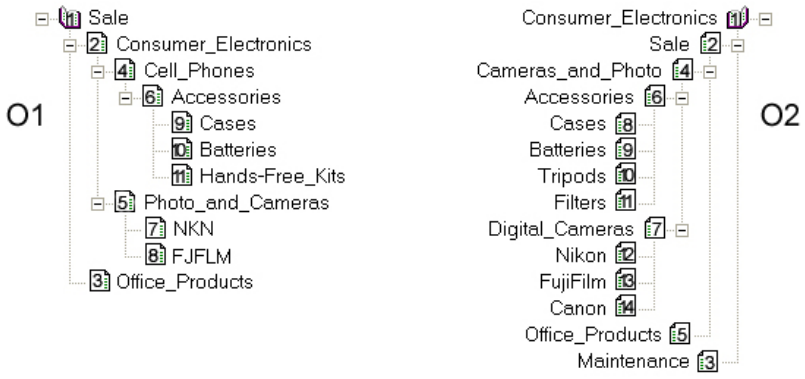

Figure 1: Two XML schemas

2.2 Data integration

Data integration is a process of generating a global virtual ontology from multiple local sources without actually loading their data into a central warehouse [44]. Data integration allows interoperation across multiple local sources having access to up-to-date data.

The scenario is as follows. First, local information sources participating in the application, *e.g.*, bookstores and cultural heritage, are identified. Then, a virtual common ontology is built. Queries are posed over the virtual common ontology, and are then reformulated into queries over the local information sources. For instance, in the e-commerce example of Figure 1, integration can be achieved by generating a single global ontology to which queries will be submitted and then translated to the local ontologies. This allows users to avoid querying the local information sources one by one, and obtain a result from them just by querying a common ontology. In order to enable semantics-preserving query answering, correspondences between semantically related entities of the local information sources and the virtual ontology are to be established, which is a matching step. Query answering is then performed by using these correspondences in the settings of Local-as-View (LAV), Global-as-View (GAV), or Global-Local-as-View (GLAV) methods [53].

2.3 Peer-to-peer information systems

Peer-to-peer is a distributed communication model in which parties (also called peers) have equivalent functional capabilities in providing each other with data and services [88]. P2P networks became popular through a file sharing paradigm, *e.g.*, music, video, and book sharing. These applications describe file contents by a simple schema (set of attributes, such as title of a song, its author, *etc.*) to which all the peers in the network have to subscribe. These schemas cannot be modified locally by a single peer.

Since peers are meant to be totally autonomous, they may use different terminologies and metadata models in order to represent their data, even if they refer to the same domain of interest [1, 48, 88]. Thus, in order to establish (meaningful) information exchange between peers, one of the steps is to identify and characterize relationships between their ontologies. This is a matching operation. Having identified the relationships between ontologies, they can be used for the purpose of query answering, *e.g.*, using techniques applied in data integration systems.

Such applications pose additional requirements on matching solutions. In P2P settings, an assumption that all the peers rely on one global schema, as in data integration, cannot be made because the global schema may need to be updated any time the system evolves [40]. While in the case of data integration schema matching can be performed at design time, in P2P applications peers need to coordinate their databases on-the-fly, therefore ultimately requiring run time schema matching.

Some P2P scenarios which rely on different types of peer ontologies, including relational schemas, XMLs, RDFs, or OWL ontologies are described in [10, 88, 48, 64, 76]. It is worth noting that most of the P2P data management projects, including [2] as well as Piazza [48] and Hyperion [75], focus on various issues of query answering and assume that the correspondences between peer schemas have been determined beforehand, and, hence, can be used for query propagation and rewriting.

2.4 Web service composition

Web services are processes that expose their interface to the web so that users can invoke them. Semantic web services provide a richer and more precise way to describe the services through the use of knowledge representation languages and ontologies. Web service discovery and integration is the process of finding a web service that can deliver a particular service and composing several services in order to achieve a particular goal, see [68, 67, 36, 32]. However, semantic web service descriptions do not necessarily reference the same ontology. Henceforth, both for finding the adequate service and for interfacing services it is necessary to establish the correspondences between the terms of the descriptions. This can be provided through matching the corresponding ontologies. For example, a browsing service may provide its output description using ontology $O1$ of Figure 1 while a purchasing service may use ontology $O2$ for describing its input. Matching ontologies is used in this context for (i) checking that what is delivered by the first service matches what is expected by the second one, (ii) verifying preconditions of the second service, and (iii) generating a mediator able to transform the output of the first service into input of the second one [30].

2.5 Query answering on the deep web

In some of the above considered scenarios, *e.g.*, schema integration and data integration, it was assumed that queries were specified by using the terminol-

ogy of a global schema. In the scenario under consideration, we discard this assumption, and therefore, users are free to pose queries by using their own terminology.

The so-called *deep web*, is made of web sites searchable via query interfaces (HTML forms) giving access to one or more back-end web databases. It is believed that it contains much more information [46] than the billions of static HTML pages of the surface web. For example, according to the investigations of [7] in March 2000, the size of the deep web was estimated as approximately from 400 to 550 times larger than the surface web. According to estimations of [46] in April of 2004, the deep web has expanded from 2000 by 3-7 times. At the moment, search engines are not very effective at crawling and indexing the deep web, since they cannot meaningfully handle the query interfaces. For example, according to [46], Google[7] and Yahoo[8] both manage to index 32% of the existing deep web objects. Finally, the deep web remains largely unexplored. However, it contains a huge number of on-line databases, which may be of use.

Thus, users have difficulties, first in discovering the relevant deep web resources and then in querying them. A standard use case includes, for example, buying a book with the lowest price among multiple on-line book stores. Query interfaces can be viewed as simple schemas (sets of terms). For example, in the book selling domain, the query interface of an on-line bookstore can be considered as a schema represented as a set of concept attributes, namely Author, Title, Subject, ISBN, Publisher. Thus, in order to enable query answering from multiple sources on the deep web, it is necessary to identify semantic correspondences between the attributes of the query interfaces of the web sites involved in handling user queries. This correspondences identification is a matching operation. Ultimately, these correspondences are used for on-the-fly translation of a user query between interfaces of web databases. For example, this motivating setup served in the basis of OntoBuilder [35], two holistic matching approaches presented in [45, 83], and others.

The above considered scenarios suggest that ontology matching is of great importance. Moreover, a need for matching is not limited to one particular application. In fact, it exists in any application involving more than one party. Thus, it is reasonable to consider ontology matching as a unified object of study. However, there are notable differences in the way these applications use matching. The application related differences must be clearly identified in order to provide the best suited solution in each case [30].

3 Basics

There have been different formalizations of matching and its result, see, for example, [11, 53, 49, 16, 81, 24, 30]. We provide here a general definition, synthesized from [21, 24, 80, 30]. In this chapter we focus on ontology matching

[7] http://www.google.com
[8] http://www.yahoo.com

and we therefore start with an informal description of what an ontology is. An *ontology* is "a specification of a conceptualization" [42], where conceptualization is an abstract view of the world represented as a set of objects. The term has been used in different research areas, including philosophy (where it was coined), artificial intelligence, information sciences, knowledge representation, object modeling, and most recently, eCommerce applications. For our purposes, an ontology can be described as a set of terms (vocabulary) associated with certain semantics and relationships. Depending on the precision of this specification, the notion of ontology encompasses several data and conceptual models, *e.g.*, classifications, database schemas, thesauri, and fully axiomatized theories. For the last model, ontologies may be represented using a Description Logic [25], where subsumption typifies the semantic relationship between terms; or Frame Logic [50], where a deductive inference system provides access to semi-structured data.

The *matching* operation determines an alignment A' (to be shortly defined) for a pair of ontologies $O1$ and $O2$. For this purpose only, we consider $O1$ and $O2$ to be finite sets of *entities*. In this general framework, we set no particular limitations on the notion of entities. Therefore, entities can be both simple and compound, compound entities should not necessarily be disjoint, *etc.*

Alignments express correspondences between entities belonging to different ontologies. A correspondence expresses the two corresponding entities and the relation that is supposed to hold between them. It is formally defined as follows:

Definition 1 (Correspondence) *Given two ontologies, a correspondence is a 5-tuple:*

$$\langle id, e_1, e_2, n, R \rangle,$$

such that

- *id is a unique identifier of the given correspondence;*
- e_1 *and* e_2 *are entities (*e.g.*, tables, XML elements, properties, classes) of the first and the second ontology, respectively;*
- n *is a confidence measure (typically in the* $[0, 1]$ *range) holding for the correspondence between* e_1 *and* e_2*;*
- R *is a relation (*e.g.*, equivalence* $(=)$*, more general* $(\sqsupseteq)$*, disjointness* $(\perp)$*, overlapping* $(\sqcap)$*) holding between* e_1 *and* e_2*.*

The correspondence $\langle id, e_1, e_2, n, R \rangle$ asserts that the relation R holds between the ontology entities e_1 and e_2 with confidence n. The higher the confidence, the higher is the likelihood of the relation to hold.

Let $\mathcal{O} = O1 \times O2$ be the set of all possible *entity correspondences* between $O1$ and $O2$ (as defined in Definition 1). To demonstrate the notion of a correspondence, let us consider Figure 1. Using some matching algorithm based on linguistic and structure analysis, the confidence measure (of the equivalence

relation to hold) between entities with labels Photo_and_Cameras in $O1$ and Cameras_and_Photo in $O2$ could be 0.67. Let us suppose that this matching algorithm uses a threshold of 0.55 for determining the resulting alignment, *i.e.*, the algorithm considers all the pairs of entities with a confidence measure higher than 0.55 as correct correspondences. Thus, our hypothetical matching algorithm should return to the user the following correspondence:

$$\langle id_{5,4}, Photo\_and\_Cameras, Cameras\_and\_Photo, 0.67, = \rangle.$$

However, the relation between the same pair of entities, according to another matching algorithm which is able to determine that both entities mean the same thing, could be exactly the equivalence relation (without computing the confidence measure). Thus, returning

$$\langle id_{5,4}, Photo\_and\_Cameras, Cameras\_and\_Photo, n/a, = \rangle.$$

Definition 2 (Alignment) *Given two ontologies $O1$ and $O2$, an alignment is made up of a set of correspondences between pairs of entities belonging to $O1$ and $O2$, respectively. The power-set $\Sigma = 2^{\mathcal{O}}$ captures the set of all possible* ontology alignments *between $O1$ and $O2$.*

This definition of the matching process makes use of three matching features in addition to the input ontologies, namely: (i) alignment A, which is to be completed by the process; (ii) matching parameters, p, *e.g.*, weights and thresholds; and (iii) external resources used by the matching process, r, *e.g.*, common knowledge and domain specific thesauri.

Definition 3 (Matching process) *The matching process can be viewed as a function f which, from a pair of ontologies $O1$ and $O2$ to match, an input alignment A, a set of parameters p and a set of oracles and resources r, returns an alignment A' between these ontologies:*

$$A' = f(O1, O2, A, p, r)$$

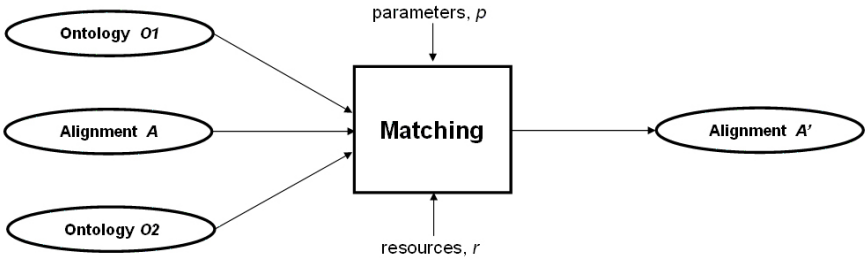

Figure 2: The matching process

The matching process can be schematically represented as illustrated in Figure 2. This definition of matching can be extended in a straightforward way to *multi-ontology matching*, that is, when multiple ontologies are taken as input. For simplicity of the presentation we focus here on matching between two ontologies.

In conceptual models and databases, the terms multiplicity or cardinality denote the constraints on a relation. Usual notations include $1:1$ (one-to-one), $1:m$ (one-to-many), $n:1$ (many-to-one) and $n:m$ (many-to-many). These naturally apply to the correspondences, thereby relating one or more entities of one ontology to one or more entities of another ontology.

Cardinality is only one (albeit important) example of a broader notion of alignment correctness. We introduce correctness into the matching process using a boolean function $\Gamma : \Sigma \to \{0, 1\}$ that captures application-specific constraints on the process, *e.g.*, cardinality constraints and correspondence constraints. In what follows, by $\Sigma_\Gamma \subseteq \Sigma$ we denote the set of all *valid* ontology alignments in Σ, that is $\Sigma_\Gamma = \{\sigma \in \Sigma \mid \Gamma(\sigma) = 1\}$. The output of the matching process is an alignment $\sigma \in \Sigma_\Gamma$, where the process may define an (either implicit or explicit) ordering over Σ, and can provide the top ranked valid alignment. Here, we also define an *exact alignment* to be a valid alignment $\sigma^* \in \Sigma_\Gamma$ that is recognized to be correct by an external observer.

In [21, 24] a 2-step method was proposed for the matching process (the rectangle in Figure 2). In the first step, a real-valued degree of similarity is automatically assigned with each correspondence. If $O1$ and $O2$ are of arity $n1$ and $n2$, respectively, then this step results in an $n1 \times n2$ *similarity matrix* M, where $M_{i,j}$ represents the degree of similarity between the i-th entity of $O1$ and j-th entity of $O2$. Various matching instantiations differ mainly in the measures of similarity they employ, yielding different similarity matrices. These measures can be arbitrarily complex, and may use various techniques for name matching, domain matching, structure matching (such as XML hierarchical representation), *etc.*

In the second step of the process, the similarity information in M is used to quantify the quality of different alignments in Σ. A single alignment is then chosen as the best alignment. The best alignment is typically considered to be the one that maximizes some *local aggregation function* (or *l-aggregator*, for short)

$$f(\sigma, M) = f(M_{1,\sigma(1)}, \ldots, M_{n,\sigma(n)}),$$

that is, a function that aggregates the degrees of similarity associated with the individual correspondences forming the alignment σ. The most common choice of l-aggregator turns out to be the sum (or equivalently, average) of correspondence degrees of similarity (*e.g.*, see [21, 56, 35]). In certain domains, however, other l-aggregators have been found appealing. For instance, an l-aggregator called *Dice* [21] stands for the ratio of the number of successfully matched correspondences (those that their similarity measure has passed a given threshold) and the total number of entities in both ontologies.

4 Ontology matching quality models

In this section we discuss issues of quality in ontology matching. We start with a brief review of the management of imperfect information, based on [19], followed by an overview of the efforts in ontology matching quality management.

4.1 Brief introduction to information imperfection

Data management tools deal regularly with imperfect information. Imperfection may be in the form of imprecision, vagueness, uncertainty, incompleteness, inconsistency, *etc.* Managing imperfections, both at the modelling (design time) level and at the querying (run time) level, can be done using tools such as probability theory, Dempster-Shafer theory, fuzzy logic, surprisal, and entropy. Over the years, several categorical classifications of the different types and sources of imperfect information have been presented. In accordance with the classifications of Bosc and Prade [15], Motro [63] and Parsons [70], imperfect information can be categorized as follows:

Uncertain information Information for which it is not possible to determine whether it is true or false.

Imprecise information Information that is not as specific as it should be.

Vague information Information that include elements (*e.g.*, predicates or quantifiers) that are inherently "vague" (in the common day-to-day sense of the word cf. [63]).

Inconsistent information Information that contains two or more assertions that cannot simultaneously hold.

Incomplete information Information for which some data are missing.

Data management approaches to deal with *uncertainty* include the possibilistic approaches and the probabilistic approaches. With possibilistic approaches, possibility theory [87] is used, where a possibility distribution is used to model the value of an attribute which is known to be uncertain. Each possible value for the attribute is assigned a membership grade that is interpreted as the degree of uncertainty [71]. Furthermore, possibility and necessity measures are attached to each result in the result set of a query. Probabilistic approaches are based on probability theory, where each result in the result set of a query is extended with a probability, representing the probability of it belonging to the set [85].

Both approaches have their advantages and disadvantages. Probabilities represent the relative occurrence of an event and therefore provide more information than possibilities. Possibilities, however, are easier to apply because they are not restricted by a stringent normalization condition of probability theory. A probabilistic approach towards ontology matching was utilized in several works, including [8, 23], where machine learning was utilized in estimating correspondence similarity measures. For example, given a correspondence

$\langle id, e_1, e_2, n, R\rangle$, the naïve Bayes method compares the probability of a set of instances of entity e_1 (e.g., brands of element NKN in $O1$, Figure 1) to serve as instances of entity e_2 (e.g., brands of entity Nikon in $O2$, Figure 1) with the probability of not serving as e_2's instances. A probability space is constructed using training data and then used for generating new correspondences.

Another work in [65], where an approach, based on combining Horn predicate logics and probability theory, was presented to harness correspondence uncertainty. A set of candidate Horn predicate rules is generated and assigned a weight. Then, a set of rules with maximum probability is selected. This work is in line with the 2-layer approach also suggested in [77] for managing uncertain data.

Imprecision of data is mostly modelled with fuzzy set theory [86] and its related possibility theory [87]. Fuzzy set theory is a generalization of regular set theory in which it is assumed that there might be elements that only partially belong to a set. Therefore, a so-called membership grade, denoting the extent to which the element belongs to the fuzzy set, is associated with each element of the universe. Two main approaches can be distinguished when modeling imprecision. First, similarity relations are utilized to model the extent to which the elements of an attribute domain may be interchanged [17]. Second, possibility distributions [71] are used, having the benefit of being suitable to cope with uncertainty (see above) and *vagueness*. In [34], a fuzzy model of ontology matching was proposed. In this model, a correspondence is assigned with a fuzzy membership degree (similar to probability, yet without a naïve assumption of correspondence independence and without constraints that stem from the need to build a probability space). Using such a model, the work continues to discuss the properties of various aggregators, transforming correspondence membership degrees into alignment similarity grades.

The treatment of *incomplete* information in databases has been widely addressed in research. A survey that gives an overview of the field is presented in [26]. The most commonly adopted technique is to model missing data with a pseudo-description, called *null*, denoting "missing" information. A more recent approach, based on possibility theory, [84] provides an explicit distinction between the cases of unknown data and inapplicable data.

4.2 Ontology matching evaluation

Quantitative quality measures for alignment evaluation, in works such as [20] consist of *precision*, *recall*, and a couple of their derivatives, namely *F-Measure* and *overall*. Assume that out of the $n_1 \times n_2$ correspondences $c \leq n_1 \times n_2$ are the correct correspondences, with respect to some reference alignment. Also, let $t \leq c$ be the number of correspondences, out of the correct correspondences, that were chosen by the matching algorithm and $f \leq n_1 \times n_2 - c$ be the number of incorrect such correspondences. Then, precision is computed to be $\frac{t}{t+f}$ and recall is computed as $\frac{t}{c}$. Clearly, higher values of both precision and recall are desired. Another derivative of precision and recall, dubbed *error*, was used in [61]. In many research works, precision and recall are considered to provide

a form of pragmatic soundness and completeness. Towards this end, an *exact alignment* is needed, against which such soundness and completeness are measured. Notice that these measures actually derive their values from a discrete domain in $[0, 1]$.

In [41], a probabilistic interpretation was assigned with precision and recall, generating posterior distributions for them. The authors have shown the benefit of such an approach to estimate the performance of text retrieval systems. However, to the best of our knowledge, such a model has never been adopted to evaluate ontology matching results so far.

A model for representing uncertainty in schema matching was presented in [34] and will be discussed in Section 4.3. Alignments were evaluated in [6] using semantic soundness and completeness. They start from some representation language $\mathcal{L}$ (*e.g.*, a Description Logic language [4]). A schema matcher α is *semantically sound* w.r.t. F if for any correspondence $\langle id, e_1, e_2, n, r\rangle$, if $\alpha(\langle e_1, e_2, r\rangle) = T$ then $\mathcal{O} \models_{\mathcal{L}} \mathcal{T}(e_1) r \mathcal{T}(e_2)$. α is *semantically complete* w.r.t. F if for any two nodes e_1 and e_2, if $\mathcal{O} \models_{\mathcal{L}} \mathcal{T}(e_1) r \mathcal{T}(e_2)$ then $\alpha(\langle e_1, e_2, r\rangle) = T$. While providing a theoretical foundation for evaluating matchers, such correctness depends on the completeness of the ontology in use. The authors use a philosophical argument of H. Putnam [72] to say that "two agents may agree at the conceptual level, but not at the pragmatic level." That is, while a matcher may correctly identify a relationship between two concepts, it may still not entail agreement at the instance level. With such an argument at hand, tasks such as query answerability, which is one of the tasks addressed in [54] by using a formal representation language, and query rewriting, which was presented as one ultimate goal of schema matching in [35], cannot be evaluated in such a framework to be sound and complete. In particular, the use of certain answers, [3] which lies heavily on the ability to agree at the conceptual level, may be hindered.

4.3 Imperfection in ontology matching

Imperfection in ontology matching has been discussed both in [54] and in [6]. The former argues for the need "to incorporate *inaccurate* mappings [correspondences] and handle *uncertainty* about mappings. Inaccuracy arises because in many contexts there is no precise mapping ... mappings may be inaccurate [since] the mapping language is too restricted to express more accurate mappings." [6] went even further, arguing philosophically that even if two ontologies fully agree on the semantics and the language is rich enough, ontologies may still not convey the same meaning, due to some hidden semantics, beyond the scope of the ontologies. A similar argument was provided in [59] in the context of relational databases: "the syntactic representation of schemas and data do not completely convey the semantics of different databases." Therefore, [54] argues that "when no accurate mapping [correspondence] exists, the issue becomes choosing the *best* mapping from the viable ones." This highlights a possible benefit of specifying semantics explicitly for the purpose of efficiently pruning the search space, to allow the evaluation of valid alignments only, namely alignments

that satisfy the semantic constraints of the model.

One way of modeling ontology matching as an uncertain process is to use similarity matrices as a measure of certainty. This way, a matcher needs to be measured by the fit of its estimation of a certainty of a correspondence to the real world. In [34], such a formal framework was provided, attempting to answer the question of whether there are "good" and "bad" matchers.

We have already observed that precision (denoted as $p(\sigma)$ for any alignment $\sigma \in \Gamma$) takes its values from a discrete domain in $[0, 1]$. Therefore, one can create equivalence alignment classes on Γ. Two alignments σ' and σ'' belong to a class p if $p(\sigma') = p(\sigma'') = p$, where $p \in [0, 1]$. Let us consider now two alignments, σ' and σ'', such that $p(\sigma') < p(\sigma'')$. For each of these two alignments we can compute their level of certainty, $f(\sigma', M)$ and $f(\sigma'', M)$, respectively. We say that a matcher is *monotonic* if for any two such alignments $p(\sigma') < p(\sigma'') \rightarrow f(\sigma', M) < f(\sigma'', M)$. As an example, consider once more Figure 1 and take two alignments, σ and σ', that differ on a single correspondence. In σ, NKN is matched to Nikon, while in σ', NKN is matched to FujiFilm. Clearly, the former is a correct correspondence while the latter is not. Therefore, $p(\sigma) < p(\sigma')$. If a matcher is monotonic, it should generate a similarity matrix M such that $f(\sigma, M) < f(\sigma', M)$.

A monotonic ontology matcher can easily identify the exact alignment. Let σ^* be the exact alignment, then $p(\sigma^*) = 1$. For any other alignment σ', $p(\sigma') \leq p(\sigma^*)$, since p takes its values in $[0, 1]$. Therefore, if $p(\sigma') < p(\sigma^*)$ then from monotonicity $f(\sigma', M) < f(\sigma^*, M)$. All one has to do then is to devise a method for finding an alignment σ that maximizes f.[9] In fact, this is one of the two most common methods for identifying the exact alignments nowadays [21, 34, 14]. The other common method, adopted in [56, 45] and others, is to only determine M automatically, allowing the user to identify the exact (ontology) alignment from the individual correspondences.

4.4 Imperfection as an emergent semantics

Imperfection can be managed and reduced using an iterative process. In such a process, initial assumptions are strengthened or discarded, and initial measures of imperfection are being refined. Such an iterative process may involve bringing together and relating information located at different places. Alternatively, one may attempt accessing a user with well-defined questions that eventually will minimize imperfection. In approaches based on possibility theory refinement can be done by composing all available fuzzy sets related to the same imperfect data. Hereby, the intersection operators for fuzzy sets (t-norms) can be used as composition operators [87].

As an example to the latter, in [33] uncertainty is refined by a comparison of K alignments, each with its own uncertainty measure (modeled as a fuzzy relation over the two ontologies). The process yields an improved ontology

[9] In [34] it was shown that while such a method works well for fuzzy aggregators (*e.g.*, weighted average) it does not work for t-norms such as min.

matching, with higher precision. For example, assume that the second-best correspondence, as generated by some heuristic, changes a correspondence of NKN with Canon to Nikon. The latter correspondence then remains unchanged for the next eight best correspondences. Therefore, in nine out of the top-10 correspondences, the correspondence of NKN with Nikon exists. If we set a threshold of 9, requiring a correspondence to appear in at least nine out of ten correspondences, then this correspondence will be included in the final alignment.

5 Matching ensembles

Striving to increase robustness in the face of the biases and shortcomings of individual matchers, tools combine principles by which different ontology matchers judge the similarity between concepts. The idea is appealing since an ensemble of complementary matchers can potentially compensate for the weaknesses of each other. Another argument in favor of ensembling was presented in [13, 74, 52]. There, ensembling was promoted as a method for ensuring matching system extensibility. Indeed, several studies report on encouraging results when using matcher ensembles (*e.g.*, see [21, 29, 35, 55, 13, 62]).

Formally, let us consider a set of m matchers $matcher_1, \ldots, matcher_m$, utilizing (possibly different) local aggregators $f^{(1)}, \ldots, f^{(m)}$, respectively. Given two ontologies $O1$ and $O2$ as before, these matchers produce an $m \times n1 \times n2$ similarity cube of $n1 \times n2$ similarity matrices $M^{(1)}, \ldots, M^{(m)}$. In these matrices, $M^{(l)}_{i,j}$ captures the degree of similarity that $matcher_l$ associates with correspondence of the i-th entity of $O1$ to the j-th entity of $O2$.

Given such a set of matchers $matcher_1, \ldots, matcher_m$, we would like to aggregate the similarity measures, given the correspondences produced by the different matchers. Such a weight aggregation can be modeled using a *global aggregation function* (or *g-aggregator*, for short) $F\left(f^{(1)}(\sigma, M^{(1)}), \cdots, f^{(m)}(\sigma, M^{(m)})\right)$. For instance, a natural candidate for g-aggregator would be as follows:

$$F\left(f^{(1)}(\sigma, M^{(1)}), \cdots, f^{(m)}(\sigma, M^{(m)})\right) = \frac{\lambda}{m} \sum_{l=1}^{m} k_l f^{(l)}(\sigma, M^{(l)})$$

It is interpreted as a (weighted) sum (with $\lambda = m$) or a (weighted) average (with $\lambda = 1$) of the local similarity measures, where k_l are some arbitrary weighting parameters.

COMA [21], which introduced first the notion of a similarity cube reverses the roles of local and global aggregators. It first reduces the cube into a matrix, and then applies to this matrix the (common) local aggregator. Many other tools (with the exception of OntoBuilder) implicitly follow COMA's footsteps, aggregating correspondence values before determining an alignment. In [24], the limitations of replacing global and local aggregators were discussed, mainly in the scope of generating top-K alignments.

6 Matcher self-tuning

The work in [20] specifies manual effort as a comparison criteria for measuring matchers. The discussion separates pre-match efforts from post-match efforts. The former includes training of matchers, parameter configuration, and specification of auxiliary information. The latter involves the identification of false positives and false negatives. The authors comment that "[un]fortanutely, the effort associated with such manual pre-match and post-match operations varies heavily with the background knowledge and cognitive abilities of users."

Clearly, one of the goals of ontology matching is to reduce this effort. Attempts to reduce post-match efforts focus on the generation of matchers that produce better alignments. Pre-match efforts focus on automatic parameter tuning. In this section we focus on the latter. Before dwelling into tuning, it is worthwhile mentioning here that another interesting aspect of the problem involves feature selection.

A general problem of pre-match effort was defined in [78] as follows: "Given a schema S, how to tune a matching system M so that it achieves high accuracy when we subsequently apply it to match S with other schemas." The various tuning parameters are called "knobs" in [78] and searching for the right knob values may be an intractable process. Let us first discuss a few alternatives for parameter tuning, followed by a discussion of methods to increase the efficiency of self-tuning.

An immediate approach to parameter tuning is that of machine learning. Using this approach, one provides a set of examples (positive, negative, or both) from which a tuning configuration is selected such that it optimizes a goal function. With such a configuration at hand, matching is performed. As an example, consider the LSD algorithm [23]. The algorithm uses an ensemble of learners, whose grades are combined using weighted average. To determine the weights of different learners, a linear regression is performed, aiming at minimizing the square error of the decision made by the ensemble over the test data.

Machine learning was also used in APFEL [28]. In this work users were first given the alignments for validation. Using user validation, new hypotheses were generated by APFEL and weighted using the initial feedback. User feedback was also adopted in eTuner as an additional source of information for the tuning process.

www.cybersuitors.com	www.date.com
select: Country: (cboCountries)	select: Select your Country (countrycode)
select: Birthday: (cboDays)	select: Date of Birth (dob_day)
select: Birthday: (cboMonths)	select: Date of Birth (dob_month)
select: Birthday: (cboYears)	select: Date of Birth (dob_year)
checkbox: (chkAgreement2)	image: ()
checkbox: (chkAgreement1)	checkbox: Date.com - Join Now for Free! (over18)
select: State (if in USA): (cboUSstates)	select: I am a (i_am)

Table 1: Best alignment for two "matchmaking" web sites

Another approach to tuning can be dubbed "dynamic tuning." According to this approach, knobs are not determined apriori but are rather derived from a heuristic at hand. An example of such an approach is available in [33]. For illustration purposes, we follow the example of query answering on the deep web, given in [33]. Let us consider two web sites that offer "matchmaking" services. In each of these sites, one has to fill in personal information (*e.g.*, name, country of residence, birthdate attributes). A matching algorithm called *Combined*, which is part of the toolkit of OntoBuilder [35], was applied. The algorithm returned the best alignment, containing a set of possible correspondences. A sample list of such correspondences is shown in Table 1. Each column in the table contains information about one field in a registration form in one of the web sites. The information consists of the type of field (*e.g.*, select field and checkbox), the label as appears at the web site, and the name of the field, given here in parentheses and hidden from the user. Each row in the table represents attribute correspondence, as proposed by this algorithm. The top part of the table contains four correct correspondences. The bottom part of the table contains three incorrect correspondences.

Matching algorithms face two obstacles in providing the best alignments. First, correct alignments should be identified and provided to the user. Second, incorrect alignments should be avoided. Separating correct from incorrect alignments is a hard task. When using a best alignment approach, an algorithm can discard attribute correspondences that do not reach some predefined threshold, assuming that those attribute correspondences with low similarity measures are less adequate than those with high similarity measures. By doing so, an algorithm (hopefully) increases precision, at the expense of recall. Using a threshold, however, works only in clear-cut scenarios. Moreover, tuning the threshold becomes an art in itself. As an example, let us consider Table 1. The four correct attribute correspondences received similarity measures in the range $(0.49, 0.7)$ while the other similarity measures ranged from 0 to 0.5. Any arbitrary apriori selection of a threshold may yield false negatives (if the threshold is set above 0.49) or false positives, in case the threshold is set below 0.49.

Consider now an alternative, in which the algorithm generates top-10 correspondences, that is, the best 10 correspondences between the two schemas, such that correspondence i differs from correspondences $1, 2, \ldots i-1$ by at least one attribute correspondence. For example, the second best correspondences include: (i) checkbox: (chkAgreement2) and checkbox: Date.com - Join Now for Free! (over18) as well as (ii) checkbox: (chkAgreement1) and image: () (this last attribute is actually a button and has no associated label or field name).

Stability analysis of the method proposed in [33] assumes that such a scenario represents a "shaky" confidence in this correspondence to start with and removes it from the set of proposed attribute correspondences. Simultaneous analysis of the top-10 correspondences reveals that the four correct attribute correspondences did not change throughout the 10 correspondences, while the other attributes were matched with different attributes in different correspondences. Stability analysis suggests that the four correspondences, for which consistent attribute correspondences were observed in the top-10 correspondences,

should be proposed as the "best alignment" yielding a precision of 100% without adversely affecting recall.

Tuning may be a costly effort. Exhaustive evaluation of the search space may be infeasible if tuning parameters take their values from continuous domains, and intractable even if all parameter domains are discrete. Therefore, efforts were made to reduce the search cost. Staged tuning was proposed in [78]. There, matchers were organized in an execution tree, for which the output of lower level matchers serve as input to higher level matchers. Given a K-level tree, the staged tuning starts with optimizing each matcher at the leaf level. Then, equipped with the optimal setting of the individual matchers it moves on to optimize the next level matcher, and so on and so forth.

For the tuning process to work well, there is a need of some ground truth regarding alignments. The quality of the training set has a crucial impact on the success of the tuning. In the early days of ontology matching research, the lack of an exact alignment yielded a poor validation process, in which heuristics were measured based on a few ontologies only. To alleviate this problem, two main directions were taken. The first approach, taken within the OntoBuilder project, involves a continuous effort to gather exact alignments (in the time of writing this chapter, there are over 200 exact alignments). This process is tedious and error prone, yet it provides a variety of ontologies practitioners are likely to access. The second approach, taken within the framework of the eTuner project [52] and also suggested in [51], involves the synthetic generation of a sufficient number of schema "mutations" from a few known exact alignments to allow effective learning. This approach overcomes the possible erroneous correspondences in a manually generated exact alignment. However, the quality of the learning set becomes dependent on the quality of the mutation rules. In addition, the strong correlation between mutated instances may generate biases in the learning process.

Combining the two approaches may provide a robust solution to the training set problem. In fact, a more varied training set could overcome the correlation problem, while the synthetic mutation would allow a tighter control over the learning process.

7 Conclusions

In this chapter we have introduced recent advances in ontology matching. In particular, after a brief introduction to the problem we have discussed several contemporary applications that motivate the research into automatic ontology matching as opposed to manual labor intensive effort. We have then provided a generic model of ontology matching as well as some technical details of several research directions, whose importance is highlighted by the need for automatic matching. These include the issues in matching quality, matching ensembles, and matcher self-tuning. While being far from exhaustive, we have striven to provide a good coverage of the performed efforts in these three directions. Much work is yet need to be done in these directions, including:

Ontology meta-matching: Following the model of uncertainty in ontology matching and our discussion of the usefulness of ensembles, a possible next step involves ontology meta-matching. That is a framework for composing an arbitrary ensemble of ontology matchers, and generating a list of best-ranked ontology alignments. We can formulate our task of identifying a top-K consensus ranking as an optimization problem, in which we aim at minimizing the amount of effort (in terms of time or number of iterations) the ensemble invests in identifying top alignments. Algorithms for generating a consensus ranking may adopt standard techniques for general quantitative rank aggregation and build on top of them, as proposed for example in [24].

Matcher self-tuning: This direction is still largely unexplored. In dynamic settings, such as the web, it is natural that applications are constantly changing their characteristics. Therefore, approaches that attempt to tune and adapt automatically matching solutions to the settings in which an application operates are of high importance. In particular, the challenge is to be able to perform matcher self-tuning at run time, and therefore, efficiency of the matcher configuration search strategies becomes crucial. Moreover, the configuration space can be arbitrary large, thus, searching it exhaustively may be infeasible.

Ontology matching evaluation: The evaluation of ontology matching approaches is still in its infancy. Initial steps have already been done in this direction, for example, the Ontology Alignment Evaluation Initiative (OAEI).[10] However, there are many issues to be addressed along the ontology matching evaluation lines in order to empirically prove the matching technology to be mature and reliable, including (*i*) design of extensive experiments across different domains with multiple test cases from each domain as well as new, difficult to match, and large real world test sets, (*ii*) more accurate evaluation measures, involving user-related measures, and (*iii*) automating acquisition of reference alignments, especially for large applications.

We have outlined three promising future research directions along the lines of the key themes discussed in this chapter. However, it is worth notice that ontology matching certainly requires further developments in a number of other important directions as well, including: background knowledge in ontology matching [38], social and collaborative ontology matching [89], performance and usability of matching approaches [13, 12], and infrastructures [35, 57].

Acknowledgements

Avigdor Gal has been partially supported by Technion V.P.R. Fund and the Fund for the Promotion of Research at the Technion. Pavel Shvaiko has been

[10] `http://oaei.ontologymatching.org/`

partly supported by the Knowledge Web European network of excellence (IST-2004-507482). We are very grateful to Fausto Giunchiglia, Mikalai Yatskevich and Jérôme Euzenat for many fruitful discussions on various ontology matching themes.

References

[1] K. Aberer. Guest editor's introduction. *SIGMOD Record*, 32(3):21–22, 2003.

[2] K. Aberer, P. Cudré-Mauroux, and M. Hauswirth. Start making sense: The chatty web approach for global semantic agreements. *Journal of Web Semantics*, 1(1):89–114, 2003.

[3] S. Abiteboul and O. Duschka. Complexity of answering queries using materialized views. In *Proceedings of the 17th Symposium on Principles of Database Systems (PODS)*, pages 254–263, Seattle, USA, 1998.

[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

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

[6] M. Benerecetti, P. Bouquet, and S. Zanobini. Soundness of schema matching methods. In *Proceedings of the 2nd European Semantic Web Conference (ESWC)*, pages 211–225, Hersounisous, Greece, 2005.

[7] M. Bergman. The deep web: surfacing hidden value. *The Journal of Electronic Publishing*, 7(1), 2001.

[8] J. Berlin and A. Motro. Autoplex: Automated discovery of content for virtual databases. In *Proceedings of the 9th International Conference on Cooperative Information Systems (CoopIS)*, pages 108–122, Trento, Italy, 2001.

[9] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[10] P. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data management for peer-to-peer computing: A vision. In *Proceedings of the 5th International Workshop on the Web and Databases (WebDB)*, pages 89–94, Madison, USA, 2002.

[11] P. Bernstein, A. Halevy, and R. Pottinger. A vision of management of complex models. *SIGMOD Record*, 29(4):55–63, 2000.

[12] P. Bernstein, S. Melnik, and J. Churchill. Incremental schema matching. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pages 1167–1170, Seoul, South Korea, 2006.

[13] P. Bernstein, S. Melnik, M. Petropoulos, and C. Quix. Industrial-strength schema matching. *SIGMOD Record*, 33(4):38–43, 2004.

[14] A. Bilke and F. Naumann. Schema matching using duplicates. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, pages 69–80, Tokyo, Japan, 2005.

[15] P. Bosc and H. Prade. An introduction to fuzzy set and possibility theory based approaches to the treatment of uncertainty and imprecision in database management systems. In *Proceedings of the 2nd Workshop on Uncertainty Management in Information Systems: From Needs to Solutions*, pages 44–70, Santa Catalina, USA, 1993.

[16] P. Bouquet, M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krötzsch, L. Serafini, G. Stamou, Y. Sure, and S. Tessaris. Specification of a common framework for characterizing alignment. Deliverable D2.2.1, Knowledge web NoE, 2004.

[17] B. Buckles and F. Petry. Generalised database and information systems. In J. Bezdek, editor, *Analysis of fuzzy Information*. CRC Press, 1987.

[18] B. Convent. Unsolvable problems related to the view integration approach. In *Proceedings of the 1st International Conference on Database Theory (ICDT)*, pages 141–156, Rome, Italy, 1986.

[19] P. Cudré-Mauroux. *Emergent semantics: rethinking interoperability for large scale decentralized information systems*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2006.

[20] H.-H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proceedings of the 2nd Workshop on Web, Web-Services, and Database Systems*, pages 221–237, Erfurt, Germany, 2002.

[21] H.-H. Do and E. Rahm. COMA – a system for flexible combination of schema matching approaches. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB)*, pages 610–621, Hong Kong, China, 2002.

[22] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proceedings of the 11th International Conference on World Wide Web (WWW)*, pages 662–673, Honolulu, USA, 2002.

[23] A.-H. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proceedings of the 20th International Conference on Management of Data (SIGMOD)*, pages 509–520, Santa Barbara, USA, 2001.

[24] C. Domshlak, A. Gal, and H. Roitman. Rank aggregation for automatic schema matching. *IEEE Transactions on Knowledge and Data Engineering*, 2007. forthcoming.

[25] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logic. In G. Brewka, editor, *Principles on Knowledge Representation, Studies in Logic, Languages and Information*, pages 193–238. CSLI Publications, 1996.

[26] C. Dyreson. A bibliography on uncertainty management in information systems. In A. Motro and P. Smets, editors, *Uncertainty Management in Information Systems: From Needs to Solutions*, pages 415–458. Kluwer Academic Publishers, Boston, USA, 1996.

[27] M. Ehrig and S. Staab. QOM – quick ontology mapping. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pages 683–697, Hiroshima, Japan, 2004.

[28] M. Ehrig, S. Staab, and Y. Sure. Bootstrapping ontology alignment methods with APFEL. In *Proceedings of the 4th International Semantic Web Conference (ISWC)*, pages 186–200, Galway, Ireland, 2005.

[29] D. Embley, D. Jackman, and L. Xu. Attribute match discovery in information integration: Exploiting multiple facets of metadata. *Journal of Brazilian Computing Society*, 8(2):32–43, 2002.

[30] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.

[31] J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI)*, pages 333–337, Valencia, Spain, 2004.

[32] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domingue. *Enabling Semantic Web Services: The Web Service Modeling Ontology*. Springer, 2007.

[33] A. Gal. Managing uncertainty in schema matching with top-K schema mappings. *Journal of Data Semantics*, 6:90–114, 2006.

[34] A. Gal, A. Anaby-Tavor, A. Trombetta, and D. Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *VLDB Journal*, 14(1):50–67, 2005.

[35] A. Gal, G. Modica, H. Jamil, and A. Eyal. Automatic ontology matching using application semantics. *AI Magazine*, 26(1):21–32, 2005.

[36] F. Giunchiglia, F. McNeill, and M. Yatskevich. Web service composition via semantic matching of interaction specifications. Technical Report DIT-06-080, University of Trento, Italy, 2006.

[37] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In *Proceedings of the 13rd International Conference on Cooperative Information Systems (CoopIS)*, pages 347–365, Agia Napa, Cyprus, 2005.

[38] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Discovering missing background knowledge in ontology matching. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, pages 382–386, Riva del Garda, Italy, 2006.

[39] F. Giunchiglia, M. Yatskevich, and P. Shvaiko. Semantic matching: Algorithms and implementation. *Journal on Data Semantics*, 9:1–38, 2007.

[40] F. Giunchiglia and I. Zaihrayeu. Making peer databases interact - a vision for an architecture supporting data coordination. In *Proceedings of the 6th International Workshop on Cooperative Information Agents (CIA)*, pages 18–35, Madrid, Spain, 2002.

[41] C. Goutte and É. Gaussier. A probabilistic interpretation of precision, recall and *f*-score, with implication for evaluation. In *Proceedings of the 27th European Conference on Advances in Information Retrieval Research (ECIR)*, pages 345–359, Santiago de Compostela, Spain, 2005.

[42] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[43] L. Haas, M. Hernández, H. Ho, L. Popa, and M. Roth. Clio grows up: from research prototype to industrial tool. In *Proceedings of the 24th International Conference on Management of Data (SIGMOD)*, pages 805–810, Baltimore, USA, 2005.

[44] A. Halevy, N. Ashish, D. Bitton, M. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka. Enterprise information integration: successes, challenges and controversies. In *Proceedings of the 24th International Conference on Management of Data (SIGMOD)*, pages 778–787, Baltimore, USA, 2005.

[45] B. He and K. Chang. Making holistic schema matching robust: An ensemble approach. In *Proceedings of the 11th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 429–438, Chicago, USA, 2005.

[46] B. He, M. Patel, Z. Zhang, and K. Chang. Accessing the deep web: a survey. *Communications of the ACM*, 50(5):94–101, 2007.

[47] R. Hull. Managing semantic heterogeneity in databases: a theoretical prospective. In *Proceedings of the 16th Symposium on Principles of Database Systems (PODS)*, pages 51–61, Tucson, USA, 1997.

[48] Z. Ives, A. Halevy, P. Mork, and I. Tatarinov. Piazza: mediation and integration infrastructure for semantic web data. *Jornal of Web Semantics*, 1(2):155–175, 2004.

[49] Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.

[50] M. Kifer, G. Lausen, and J. Wu. Logical foundation of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843, 1995.

[51] G. Koifman. Multi-agent negotiation over database-based information goods. Master's thesis, Technion-Israel Institute of Technology, February 2004.

[52] Y. Lee, M. Sayyadian, A. Doan, and A. Rosenthal. eTuner: tuning schema matching software using synthetic scenarios. *VLDB Journal*, 16(1):97–122, 2007.

[53] M. Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the 21st Symposium on Principles of Database Systems (PODS)*, pages 233–246, Madison, USA, 2002.

[54] J. Madhavan, P. Bernstein, P. Domingos, and A. Halevy. Representing and reasoning about mappings between domain models. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, pages 122–133, Edmonton, Canada, 2002.

[55] J. Madhavan, P. Bernstein, and E. Rahm. Generic schema matching with Cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB)*, pages 48–58, Rome, Italy, 2001.

[56] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: a versatile graph matching algorithm. In *Proceedings of the 18th International Conference on Data Engineering (ICDE)*, pages 117–128, San Jose, USA, 2002.

[57] S. Melnik, E. Rahm, and P. Bernstein. Developing metadata-intensive applications with Rondo. *Journal of Web Semantics*, 1(1):47–74, 2003.

[58] S. Melnik, E. Rahm, and P. Bernstein. Rondo: A programming platform for model management. In *Proceedings of the 22nd International Conference on Management of Data (SIGMOD)*, pages 193–204, San Diego, USA, 2003.

[59] R. Miller, L. Haas, and M. Hernández. Schema mapping as query discovery. In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB)*, pages 77–88, Cairo, Egypt, 2000.

[60] R. Miller, M. Hernàndez, L. Haas, L.-L. Yan, C. Ho, R. Fagin, and L. Popa. The Clio project: Managing heterogeneity. *SIGMOD Record*, 30(1):78–83, 2001.

[61] G. Modica, A. Gal, and H. Jamil. The use of machine-generated ontologies in dynamic information seeking. In *Proceedings of the 9th International Conference on Cooperative Information Systems (CoopIS)*, pages 433–448, Trento, Italy, 2001.

[62] P. Mork, A. Rosenthal, L. Seligman, J. Korb, and K. Samuel. Integration workbench: Integrating schema integration tools. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE) Workshops*, page 3, Atlanta, USA, 2006.

[63] A. Motro. Management of uncertainty in database systems. In W. Kim, editor, *Modern Database Systems, The object model, interoperability and beyond*. Addison-Wesley, Reading, Massachusetts, 1995.

[64] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. Edutella: A P2P networking infrastructure based on RDF. In *Proceedings of the 11th International World Wide Web Conference (WWW)*, pages 604–615, Honolulu, USA, 2002.

[65] H. Nottelmann and U. Straccia. A probabilistic, logic-based framework for automated web directory alignment. In Z. Ma, editor, *Soft Computing in Ontologies and the Semantic Web*, volume 204 of *Studies in Fuzziness and Soft Computing*, pages 47–77. Springer, 2006.

[66] N. Noy and M. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[67] S. Oundhakar, K. Verma, K. Sivashanugam, A. Sheth, and J. Miller. Discovery of web services in a multi-ontology and federated registry environment. *International Journal of Web Services Research*, 2(3):1–32, 2005.

[68] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web services capabilities. In *Proceedings of the 1st International Semantic Web Conference (ISWC)*, pages 333–347, Chia Laguna, Italy, 2002.

[69] C. Parent and S. Spaccapietra. Issues and approaches of database integration. *Communications of the ACM*, 41(5):166–178, 1998.

[70] S. Parsons. Current approaches to handling imperfect information in data and knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 8(3):353–372, 1996.

[71] H. Prade and C. Testemale. Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information Sciences*, 34:115–143, 1984.

[72] H. Putnam, editor. *Reason, Truth, and History*. Cambridge University Press, 1981.

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

[74] E. Rahm, H.-H. Do, and S. Maßmann. Matching large XML schemas. *SIGMOD Record*, 33(4):26–31, 2004.

[75] P. Rodríguez-Gianolli, M. Garzetti, L. Jiang, A. Kementsietsidis, I. Kiringa, M. Masud, R. J. Miller, and J. Mylopoulos. Data sharing in the Hyperion peer database system. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 1291–1294, Seoul, South Korea, 2005.

[76] M.-C. Rousset, P. Adjiman, P. Chatalic, F. Goasdoué, and L. Simon. Somewhere in the semantic web. In *Proceedings of the 32nd International Conference on Current Trends in Theory and Practice of Computer Science (SofSem)*, pages 84–99, Merin, Czech Republic, 2006.

[77] A. D. Sarma, O. Benjelloun, A. Y. Halevy, and J. Widom. Working models for uncertain data. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE)*, page 7, Atlanta, USA, 2006.

[78] M. Sayyadian, Y. Lee, A.-H. Doan, and A. Rosenthal. Tuning schema matching software using synthetic scenarios. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 994–1005, Trondheim, Norway, 2005.

[79] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.

[80] P. Shvaiko. *Iterative Schema-based Semantic Matching*. PhD thesis, International Doctorate School in Information and Communication Technology, University of Trento, Trento, Italy, November 2006.

[81] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal of Data Semantics*, 4:146 – 171, 2005.

[82] B. Srivastava and J. Koehler. Web service composition - Current solutions and open problems. In *Proceedings of the Workshop on Planning for Web Services at the 13th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 28–35, Trento, Italy, 2003.

[83] W. Su, J. Wang, and F. Lochovsky. Holistic schema matching for web query interfaces. In *Proceedings of the 10th Conference on Extending Database Technology (EDBT)*, pages 77–94, Munich, Germany, 2006.

[84] D. Tsichritzis and A. C. Klug. The ansi/x3/sparc dbms framework report of the study group on dabatase management systems. *Information Systems*, 3(3):173–191, 1978.

[85] S. Wong, Y. Xiang, and X. Nie. Representation of bayesian networks as relational databases. In *Proceedings of the 5th International Conference on Information Processing and Management of Uncertainty (IPMU)*, pages 159–165, Paris, France, 1994.

[86] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[87] L. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[88] I. Zaihrayeu. *Towards Peer-to-Peer Information Management Systems.* PhD thesis, International Doctorate School in Information and Communication Technology, University of Trento, Italy, March 2006.

[89] A. Zhdanova and P. Shvaiko. Community-driven ontology matching. In *Proceedings of the 3rd European Semantic Web Conference (ESWC)*, pages 34–49, Budva, Montenegro, 2006.