Chapter 4
Classifications of Ontology Matching Techniques

Having defined what the matching problem and the process for solving it are, and before scrutinising further the details of matching techniques, we classify them from different standpoints. This should help better understanding these systems.

The major contributions of the previous decades are presented in (Larson et al. 1989; Batini et al. 1986; Kashyap and Sheth 1996; Parent and Spaccapietra 1998). Later, the topic has been surveyed in (Rahm and Bernstein 2001; Wache et al. 2001; Kalfoglou and Schorlemmer 2003b). The topic was also treated in the context of data integration or ontology change (Choi et al. 2006; Flouris et al. 2008; Bellahsene et al. 2011; Doan et al. 2012). These addressed the matching problem from different perspectives (artificial intelligence, information systems, databases) and analysed disjoint sets of systems. (Shvaiko and Euzenat 2005) considered the above mentioned works together, focussing on schema-based matching methods, and aiming to provide a common conceptual basis for their analysis. Here, we follow and extend this work on classifying matching approaches.

In this chapter, we first consider various dimensions along which a classification may be designed (Sect. 4.1). We then present our classification based on several of these dimensions (Sect. 4.2) and the classes of matching techniques (Sect. 4.3). Finally, we discuss some alternative classifications of matching approaches that have been proposed so far (Sect. 4.4).

4.1 Matching Dimensions

There are many independent dimensions along which algorithms may be classified. Following the definition of the matching process in Fig. 2.8, we may primarily classify algorithms according to (i) the input of the algorithms, (ii) the characteristics of the matching process, and (iii) the output of the algorithms. The other characteristics, such as parameters, resources, and input alignments, are considered less important. Let us discuss these three main aspects in turn.

J. Euzenat, P. Shvaiko, *Ontology Matching*, DOI 10.1007/978-3-642-38721-0_4,

4.1.1 Input Dimensions

These dimensions concern the kind of input on which algorithms operate. As a first dimension, algorithms can be classified depending on the data or conceptual models in which ontologies are expressed. For example, the Artemis system (Sect. 8.1.6) supports the relational, object-oriented, and entity–relationship models; Cupid (Sect. 8.1.11) supports XML and relational models; QOM (Sect. 8.3.3) supports RDF and OWL models. A second possible dimension depends on the kind of data that the algorithms exploit: different approaches exploit different information in the input ontologies. Some of them rely only on schema-level information, e.g., Cupid (Sect. 8.1.11), COMA (Sect. 8.1.12); others rely only on instance data, e.g., GLUE (Sect. 8.2.5); and others exploit both schema- and instance-level information, e.g., QOM (Sect. 8.3.3). Even with the same data models, matching systems do not always use all available constructs, e.g., S-Match (Sect. 8.1.18), when dealing with attributes, discards information about data types and uses only the attributes names. Some algorithms focus on the labels assigned to the entities, some consider their internal structure and the types of their attributes, and others consider their relations with other entities (see next section for details).

More generally, we can consider the information *origin* as a dimension, on which matching is based: this information can come directly from the content of the ontologies to be matched or from relations between the ontologies and other external resources, called *context*. Hence, this origin dimension can be split into *internal* or *content-based matching* and *external* or *context-based matching*. External resources can be formal, such as other ontologies, or informal, such as a collection of pictures annotated by the ontology or a thesaurus describing the terms used in the ontology. Relations with such external resources can be explicit, e.g., there is already an alignment with an external ontology or links to pictures, or implicit, i.e., such relations have to be established.

4.1.2 Process Dimensions

A classification of the matching process could be based on its general properties, as soon as we restrict ourselves to formal algorithms. In particular, it depends on the *approximate* or *exact* nature of its computation. Exact algorithms compute the precise solution to a problem; approximate algorithms sacrifice exactness for performance (Ehrig and Sure 2004). All of the techniques discussed in the remainder of the book can be either approximate or exact. Another dimension for analysing matching algorithms is based on the way they interpret the input data. We identify two categories depending on whether the matcher considers the input intrinsically or through some semantic theory of the considered entities. We call these categories: *syntactic* vs. *semantic* and discuss them in detail in the next section.

4.1.3 Output Dimensions

Apart from the information that matching systems exploit and how they manipulate it, the other important class of dimensions concerns the form of the result that these systems produce. The form of the alignment might be of importance: is it a one-to-one alignment between the ontology entities? Has it to be a final correspondence? Is any relation suitable?

Some other significant distinctions in the output results have been indicated in (Giunchiglia and Shvaiko 2003). One dimension concerns whether systems deliver a graded answer, e.g., that the correspondence holds with 98 % confidence or 4/5 probability, or an all-or-nothing answer, e.g., that the correspondence definitely holds or not. In some approaches, correspondences between ontology entities are determined using distance measures. This is used for providing an alignment expressing equivalence between these entities. Another dimension concerns the kind of relations between entities a system can provide. Most of the systems focus on equivalence ($=$), while a few others are able to provide a more expressive result, e.g., equivalence, subsumption ($\leq$), and incompatibility ($\perp$) (Giunchiglia et al. 2004; Bouquet et al. 2003b; Hamdi et al. 2010b; Spiliopoulos et al. 2010).

In the next section, we present a classification of techniques that draws simultaneously on these criteria.

4.2 Classification of Matching Approaches

4.2.1 Methodology

To ground and ensure a comprehensive coverage for our classification we have analysed state-of-the-art approaches used for ontology matching. Chapter 8 reports a partial list of systems which have been scrutinised pointing to (some of) the most important contributions. We have used the following guidelines for building our classification:

Exhaustivity: The extension of categories dividing a particular category must cover its extension, i.e., their aggregation should give the complete extension of the category.

Disjointness: In order to have a proper tree, the categories dividing one category should be pairwise disjoint by construction.

Homogeneity: In addition, the criteria used for further dividing one category should be of the same nature, i.e., should come from the same dimension introduced in Sect. 4.1. This usually helps guarantee disjointness.

Saturation: Classes of concrete matching techniques should be as specific and discriminative as possible in order to provide a fine-grained distinction between possible alternatives. These classes have been identified following a *saturation* principle: they have been added and modified until saturation was reached, i.e., taking

into account new techniques did not require introducing new classes or modifying them.

Disjointness and exhaustivity of the categories ensure the stability of the classification, namely that new techniques will not occur in between two categories. Categories of matching approaches represent the state of the art. Obviously, with appearance of new techniques, they might be extended and further detailed.

The exact vs. approximate opposition has not been used because each of the methods described below can be implemented as exact or approximate algorithms, depending on the goals of the matching system.

We build on the previous work on classifying automated schema matching approaches of (Rahm and Bernstein 2001) which distinguishes between *elementary* (individual) matchers and *composition* of matchers. Elementary matchers comprise *instance-* and *schema-based*, *element-* and *structure-level*, *linguistic* and *constraint-based* matching techniques. *Cardinality* and *auxiliary information*, e.g., thesauri, global schemas, can also be taken into account.

For classifying matching techniques, we introduced two synthetic classifications in (Shvaiko and Euzenat 2005), based on what we have found to be the most salient properties of the matching dimensions. These two classifications are represented by two trees sharing their leaves. The leaves represent classes of matching techniques and their concrete examples. In this edition of this book, we revised this classification based on the evolution of the state of the art, in particular the development of new approaches, which have changed the balance with respect to the classification of the first edition, e.g., context-based matching. These two revised and updated synthetic classifications are (see Fig. 4.1):

- *Granularity/Input interpretation* classification based (i) on the matcher granularity, i.e., element- or structure-level, and then (ii) on how the techniques generally interpret the input information (Sect. 4.2.2),
- *Origin/Kind of input* classification based (i) on the origin of the information considered by the matcher, and (ii) on the kind of input taken into account by matching techniques (Sect. 4.2.3).

The overall classification of Fig. 4.1 can be read both in descending (focussing on how the techniques interpret the input information) and ascending (focussing on the origin of matching clues before their interpretation) manner in order to reach the layer of *Concrete techniques*. It is designed in a way that offers a planar graph layout.

4.2.2 Granularity/Input Interpretation *Layer*

Matchers are distinguished by the *Granularity/Input interpretation* layer according to the following classification criteria:

- *Element-level vs. structure-level*: Element-level matching techniques compute correspondences by analysing entities or instances of those entities in isolation,

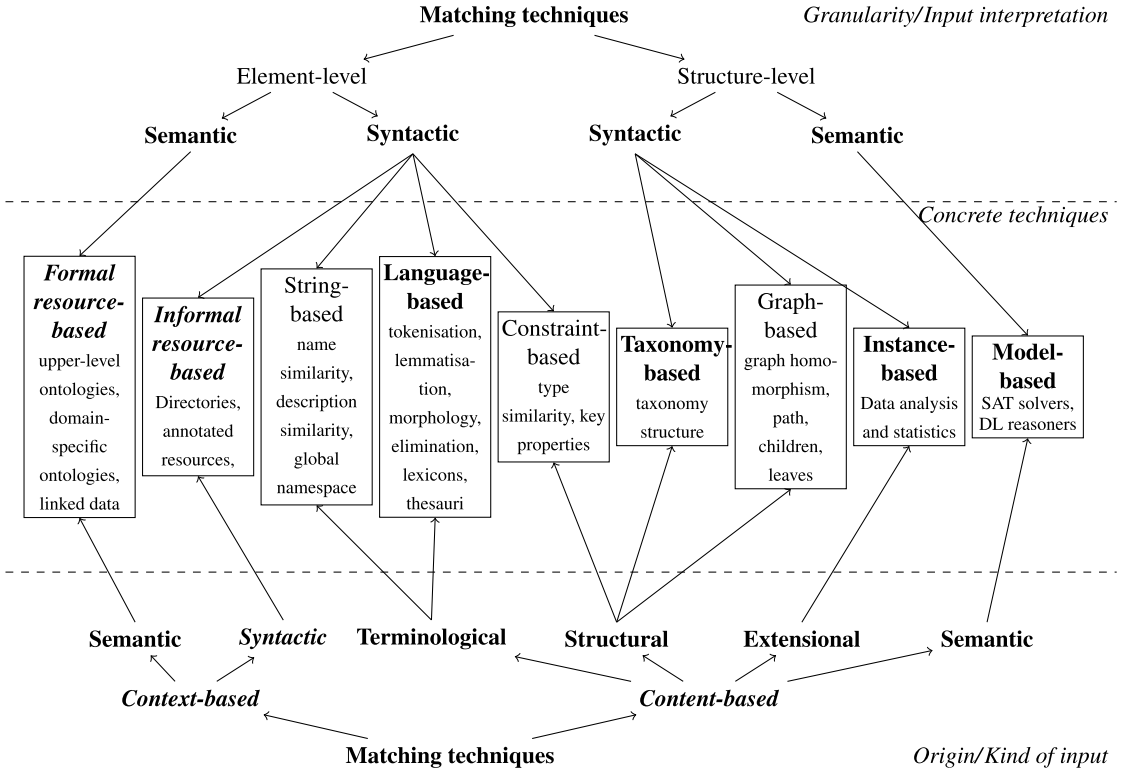

Fig. 4.1 The retained classification of matching approaches. The upper classification is based on granularity and input interpretation; the lower classification is based on the origin of information and the kind of input. The middle layer features classes of concrete matching techniques. In **bold** are new categories with respect to (Rahm and Bernstein 2001) and in ***bold-italics*** those introduced in the second edition.

ignoring their relations with other entities or their instances. Structure-level techniques compute correspondences by analysing how entities or their instances appear together in a structure. This criterion for schema-based approaches is the same as first introduced in (Rahm and Bernstein 2001), while the element-level vs. structure-level separation for instance-based approaches follows the work in (Kang and Naughton 2003).
- *Syntactic vs. semantic*: The key characteristic of the syntactic techniques is that they interpret the input with regard to its sole structure following some clearly stated algorithm. Semantic techniques use some formal semantics, e.g., model-theoretic semantics, to interpret the input and justify their results. In case of a semantic-based matching system, exact algorithms are complete with regard to the semantics, i.e., they guarantee a discovery of all the possible alignments, while approximate algorithms tend to be incomplete.

To emphasise the differences with the initial classification of (Rahm and Bernstein 2001), the new categories or classes are marked in bold. In particular, in the Granularity/Input Interpretation layer we detail further the element- and structure-level matching by introducing the syntactic vs. semantic distinction.

4.2.3 Origin/Kind of Input *Layer*

The *Origin/Kind of input* takes the origin dimension as its first level of separation, and the type of input considered by a particular technique as the second level:

- The first level is simply the content-based vs. context-based or internal vs. external distinction of the *Origin* dimension (Sect. 4.1.1).
- The second level refines these categories by distinguishing among external resources, those which are interpreted semantically, and those which are not, named *syntactic*. The *content-based matching* category is further articulated depending on which kind of data the algorithms work on strings (*terminological*), structures (*structural*), models (*semantics*) or data instances (*extensional*). The first two are found in the ontology descriptions. The third one requires some semantic interpretation of the ontology and usually uses some semantically compliant reasoner to deduce correspondences. The last one works on the actual population of an ontology. In turn, *context-based matching* is only further articulated into *syntactic* and *semantic* categories. Syntactic techniques, when considered elementary, are usually either *terminological* or *structural* or *extensional*; hence, the *informal resource-based* approaches could have been split in this three-fold way as well. However, such techniques have not been widely applied in practice so far, so we kept the presentation simpler, thus having only the *syntactic* category.

Hence, it can be considered that the *Kind of input* classification of the first edition was reduced to its first layer and that the content-based vs. context-based separation of the origin dimension was introduced on top. This follows the development of context-based matching in recent years.

4.3 Classes of Concrete Techniques

The distinctions between matching techniques in the *Concrete techniques* layer of our classification are motivated by the way in which techniques interpret the input information in each concrete case. In particular, a label can be interpreted as a string (a sequence of letters from an alphabet) or as a word or a phrase in some natural language, a hierarchy can be considered as a graph (a set of nodes related by edges) or a taxonomy (a set of concepts having a set-theoretic interpretation organised by a relation which preserves inclusion). Thus, we introduce the following classes of ontology matching techniques at the element-level: string-based, language-based, constraint-based. We also identify techniques relying on external resources related in one way or another to the ontologies to be matched. These techniques can be based on informal resources, such as text or media corpora, or formal resources, such as ontologies. At the structure-level we distinguish between graph-based, taxonomy-based, model-based, and instance-based techniques.

We discuss below the main classes of the *Concrete techniques* layer according to the above classification in more detail. Contrary to the first edition, all these classes have instances, so none of these classes are hypothetical (the hypothetical classes of the first edition have already been realised in practice). Finally, several changes have been made in this layer due to the narrowness or low representativity of some classes, namely *Alignment reuse* and *Repository of structures* are not explicitly present in Fig. 4.1 and should be considered as merged into *Formal resource-based* and *Graph-based* classes, respectively.

4.3.1 Element-Level Techniques

Element-level techniques consider ontology entities or their instances in isolation from their relations with other entities or their instances.

String-Based Techniques

String-based techniques are often used in order to match names and name descriptions of ontology entities. These techniques consider strings as sequences of letters in an alphabet. They are typically based on the following intuition: the more similar the strings, the more likely they are to denote the same concepts. Usually, distance functions map a pair of strings to a real number, such that a smaller value indicates a greater similarity between the strings. Some examples of string-based techniques that are extensively used in matching systems are prefix, suffix, edit distances, and n-gram similarity. Various such string comparison techniques are presented in Sect. 5.2.1.

Language-Based Techniques

Language-based techniques consider names as words in some natural language, e.g., English. They are based on natural language processing techniques exploiting morphological properties of the input words. Several of these techniques are presented in Sect. 5.2.2 (intrinsic techniques).

Usually, they are applied to names of entities before running string-based or lexicon-based techniques in order to improve their results. However, we consider these language-based techniques as a separate class of matching techniques since they can be naturally extended, for example, in a distance computation (by comparing the resulting strings or sets of strings).

This class now encompasses the *Linguistic resources* class of the first edition, which covered the use of linguistic resources, such as lexicons or domain-specific thesauri, to match words (in this case names of ontology entities are considered as words of a natural language) based on linguistic relations between them, e.g., synonyms, hyponyms. Several such methods are presented in Sect. 5.2.2 (extrinsic techniques). Resources, such as thesauri and lexicons, may also be used as 'ontologies' instead of linguistic resources, i.e., used to interpret concepts instead of terms. In this case, these same resources occur in *Informal resource-based techniques*.

Constraint-Based Techniques

Constraint-based techniques are algorithms that deal with the internal constraints being applied to the definitions of entities, such as types, cardinality (or multiplicity) of attributes, and keys. These techniques are presented in Sect. 5.3.

Informal Resource-Based Techniques

Ontologies may be tied to informal resources, e.g., annotating encyclopedia pages or pictures. *Informal resource-based techniques* cover techniques used for deducing relations between ontology entities based on how these are related to such resources. Typically, two classes annotating the same set of pictures can be considered equivalent. Such techniques often exploit data analysis and statistical approaches as well as approaches which take advantage of a (hopefully large) corpus of related entities to find regularities and discrepancies between them. This class of techniques covers part of the *Data analysis and statistics techniques* class of the first edition dealing with extensions (instance-based matching).

Formal Resource-Based Techniques

Formal resource-based techniques take advantage of external ontologies in order to perform matching. Usually, they compose alignments between the ontologies to

be matched originating from one or several external ontologies. Several context-based ontology matchers using formal resources have been proposed in recent years, by using domain-specific ontologies, upper-level ontologies, linked data and other resources. This class of techniques covers and generalises both the *Upper-level and domain-specific formal ontologies* and *Alignment reuse* classes of the first edition of this book. Both classes focussed on different aspects of formal resource-based techniques, namely on the kind of ontologies or on the type of alignments used as external resources. A general framework for dealing with external resource-based matching, and in particular with formal resources, is detailed in Sect. 7.3.

4.3.2 Structure-Level Techniques

Contrary to element-level techniques, structure-level techniques consider the ontology entities or their instances to compare their relations with other entities or their instances.

Graph-Based Techniques

Graph-based techniques are graph algorithms which consider the input ontologies (including database schemas, and taxonomies) as labelled graphs.

Usually, the similarity comparison between a pair of nodes from the two ontologies is based on the analysis of their positions within the graphs. The intuition behind this is that, if two nodes from two ontologies are similar, their neighbours must also be somehow similar. Different graph-based techniques are described in Sect. 6.1. Along with purely graph-based techniques, there are other more specific structure-based techniques, for instance, involving trees. Graph-based techniques cover the *Repository of structure* class of the first edition, which is now considered as part of pattern-based matching (Sect. 6.1.4).

Taxonomy-Based Techniques

Taxonomy-based techniques are also graph algorithms which consider only the specialisation relation. The intuition behind taxonomic techniques is that specialisation connect terms that are already similar (being interpreted as a subset or superset of each other), therefore their neighbours may be also somehow similar. This intuition can be exploited in several different ways presented in Sect. 6.1.

Model-Based Techniques

Model-based (or semantically grounded) algorithms handle the input based on its semantic interpretation, e.g., model-theoretic semantics. The intuition is that if two

entities are the same, then they share the same interpretations. Thus, they are well grounded deductive methods. Examples are propositional satisfiability and description logics reasoning techniques. They are further reviewed in Sect. 6.5.

Instance-Based Techniques

Instance-based techniques are those that compare sets of instances of classes in order to decide if these classes match or not. They can be based on simple set-theoretic reasoning or on more elaborate data analysis and statistical techniques. They help in grouping together items or computing distances between them. From data analysis techniques, we discuss distance-based classification, formal concept analysis and correspondence analysis; from statistical analysis methods we consider frequency distributions. Instance-based techniques are mostly described in Sect. 5.4. We exclude from this category learning techniques, which require a sample of the result, i.e., the alignment. These techniques are considered specifically in Sect. 7.5.

4.4 Other Classifications

There are some other classifications of matching techniques. For example, (Ehrig 2007) introduced a classification based on two orthogonal dimensions. These can be viewed as horizontal and vertical dimensions. The horizontal dimension includes three layers that are built one on top of another:

Data layer: This is the first layer. Matching between entities is performed here by comparing only data values of simple or complex data types.

Ontology layer: This is the second layer which, in turn, is further divided into four levels, following the 'layer cake' of (Berners-Lee et al. 2001). These are semantic nets, description logics, restrictions and rules. For example, at the level of semantic nets, ontologies are viewed as graphs with concepts and relations, and, therefore, matching is performed by comparing only these. The description logics–level brings a formal semantics account to ontologies. Matching at this level includes, for example, determining taxonomic similarity based on the number of subsumption relations separating two concepts. This level also takes into account instances of entities, therefore, for example, assessing concepts to be the same, if their instances are similar. Matching at the levels of restrictions and rules is typically based on the idea that if similar rules between entities exist, these entities can be regarded as similar.

Context layer: Finally, this layer is concerned with the practical usage of entities in the context of an application. Matching is performed here by comparing the usages of entities in ontology-based applications. One of the intuitions behind such matching methods is that similar entities are often used in similar contexts.

The vertical dimension represents specific *domain knowledge* which can be situated at any layer of the horizontal dimension. Here, the advantage of external resources

of domain-specific knowledge, e.g., Dublin Core for the bibliographic domain, is considered for assessing the similarity between entities of ontologies.

(Doan and Halevy 2005) classifies matching techniques into (i) rule-based and (ii) learning-based. Typically, rule-based techniques work with schema-level information, such as entity names, data types and structures. Some examples of rules are that two entities match if their names are similar or if they have the same number of neighbour entities. Learning-based approaches often work with instance-level information, thereby performing matching by comparing value formats and distributions of data instances underlying the entities under consideration, for example. However, learning can also be performed at schema-level and from the previous matches, e.g., as proposed in the LSD approach (Sect. 8.2.4).

(Zanobini 2006) classifies matching methods into three categories following the cognitive theory of meaning and communication between agents:

Syntactic: This category represents methods that use purely syntactic matching methods. Examples of such methods include string-based techniques and graph matching techniques.

Pragmatic: This category represents methods that rely on comparison of data instances underlying the entities under consideration in order to compute alignments. Examples of such methods include automatic classifiers, and formal concepts analysis (Sect. 5.4.1).

Conceptual: This category represents methods that work with concepts and compare their meanings in order to compute alignments. Examples of such methods include techniques exploiting external thesauri, such as WordNet (Sect. 5.2.2), in order to compare senses among the concepts under consideration.

There were also some classifications mixing the process dimension of matching together with either input dimension or output dimension. For example, (Do 2005) extends the work of (Rahm and Bernstein 2001) by adding a *reuse-oriented* category of techniques on top of schema-based vs. instance-based separation, meaning that reuse-oriented techniques can be applied at schema- and instance-level. However, these techniques can also include some input information, such as user input or alignments obtained from previous match operations.

Finally, the more the ontology matching field progresses, the wider the variety of techniques that come into use at different levels of granularity. For example, machine learning methods, which where often applied only to the instance-level information, also started being applied more widely to schema-level information. We believe that such a cross-fertilisation will gain more support in the future. Therefore, ultimately, it could be the case that any mathematical method will find appropriate uses for ontology matching.

4.5 Summary

Following the complexity of ontology definition, a variety of techniques may be used. This chapter has shown the difficulty of having a clear cut classification of

algorithms. In Sect. 4.2, we provided two such classifications based on granularity and input interpretation on the one hand and the origin and the kind of input on the other hand.

The classifications discussed in this chapter provide a common conceptual basis for organising matching techniques. They can be used for comparing (analytically) different existing ontology matching systems as well as for designing new ones, taking advantage of state-of-the-art solutions. The classifications of matching methods also provide some guidelines which help identifying families of matching approaches.

In the following three chapters we first present basic techniques (Chap. 5), which exploit local characteristics of entities, then advanced techniques (Chap. 6), which aim at considering all the characteristics of entities, thus treating them globally, and finally, strategies (Chap. 7) used to build matching systems.

http://www.springer.com/978-3-642-38720-3

Ontology Matching
Euzenat, J.; Shvaiko, P.
2013, XVII, 511 p. 103 illus., 1 illus. in color., Hardcover
ISBN: 978-3-642-38720-3