An Approach to Extracting Interschema Properties from XML Schemas at Various "Severity" Levels

Pasquale De Meo, Giovanni Quattrone and Domenico Ursino
Università Mediterranea di Reggio Calabria
Via Graziella, Località Feo di Vito
89122 Reggio Calabria, Italy
E-mail: demeo@unirc.it, quattrone@unirc.it, ursino@unirc.it

Giorgio Terracina
Dipartimento di Matematica
Università della Calabria
Via Pietro Bucci
87036 Rende (CS), Italy
E-mail: terracina@mat.unical.it

Keywords: XML schemas, synonymies, homonymies, hyponymies, overlappings, interscheme property extraction

Received: June 7, 2006

This paper presents an approach for the semi-automatic, uniform extraction of synonymies, hyponymies, overlappings and homonymies holding among concepts of different XML Schemas. The proposed approach is specialized for XML, is almost automatic and "light". As a further, original, peculiarity, it is parametric w.r.t. a "severity level" against which the extraction task is performed. First the paper presents an overview of the interschema property extraction approaches already presented in the past, as well as a set of criteria for classifying this kind of approaches. After this, it describes the proposed approach in all details, illustrates various theoretical results, presents the experiments we have performed for testing it and compares it with the interschema property extraction approaches previously proposed in the literature.

Povzetek: Opisan je polavtomatski postopek za ekstrakcijo sinonimov iz XML shem.

1 Introduction

The Web is becoming the reference infrastructure for most of the applications conceived to handle the interoperability among partners. As a matter of fact, it is presently playing a key role for both the publication and the exchange of information among organizations. In order to make Web activities easier, the World Wide Web Consortium proposed XML (eXtensible Markup Language) for unifying representation capabilities, typical of HTML, and data management features, typical of classical DBMSs.

The exploitation of XML is crucial for improving the interoperability of Web partners; as a matter of fact, this language provides a uniform format for exchanging data among them. However, XML usage alone is not enough for guaranteeing such a cooperation. In fact, the heterogeneity of data exchanged over the Web regards not only their formats but also their semantics. The use of XML allows format heterogeneity to be faced; the exploitation of XML Schemas allows the definition of a reference context for exchanged data and is a first step for handling semantic diversities; however, in order to completely and satisfactorily manage these last, the knowledge of interschema properties (see Section 2.1), possibly holding among concepts belonging to different sources, is necessary.

The most common interschema properties previously considered in the literature are synonymies and homonymies. A *synonymy* between two concepts indicates that they have the same meaning. An *homonymy* between two concepts denotes that they refer to different meanings, yet having the same name. In the past some approaches have been also proposed for deriving other interschema properties, e.g., hyponymies and overlappings. A concept C_1 is a *hyponym* of a concept C_2 (that is, in its turn, a *hypernym* of C_1) if C_1 has a more specific meaning than C_2. As an example, "PhD Student" is a hyponym of "Student". An *Overlapping* holds between two concepts if they are not synonymous but share a significant set of properties.

For a more detailed survey about the semantic relationships possibly occurring between two concepts the reader is referred to [24]. In this paper semantic relationships are defined and classified according to different perspectives and disciplines, such as linguistics, logics and cognitive psychology. From a comparison between the definitions of [24] and those introduced in this paper we can observe that: *(i)* our definition of synonymy exactly matches the definition of synonymy provided in [24]; *(ii)* our concept of homonymy can be regarded as a special case of the con-

cept of antinomy specified in [24]; specifically, in that paper, an antinomy exists between two terms if they denote opposite (or, at least, different) concepts; our definition of homonymy, instead, requires that two terms indicate different concepts and, in addition, that they share the same name; *(iii)* our hyponymy property corresponds to the inclusion relationship specified in [24]; *(iv)* our overlapping property is similar to some kinds of meronymic relationship introduced in [24] (these last indicate that a part of a concept A is someway related to a part of a concept B).

Owing to the enormous increase of the number of available information sources, all the approaches for interschema property extraction currently proposed in the literature are *semi-automatic*; specifically, they require the human intervention only during a pre-processing phase and for the validation of obtained results. The rapid development of the Web leads each interschema property extraction approach to operate on a great number of sources; this requires a further effort for conceiving approaches with less manual intervention.

Since the possible interschema properties to consider are numerous and various, the capability of *uniformly* deriving distinct properties appears to be a crucial feature for a new interschema property derivation approach. As a matter of fact, different strategies for extracting distinct interschema properties could lead to different interpretations of the same reality; this is a situation that must be avoided.

Finally, the large number of currently available information sources makes it evident the necessity that an interschema property derivation approach should be "light", i.e., it should minimize the exploitation of thresholds and/or weights whose tuning requires a lot of efforts.

This paper provides a contribution in this setting and proposes an approach for uniformly extracting synonymies, hyponymies, overlappings and homonymies from a set of XML Schemas.

Our approach satisfies all the desiderata mentioned above. In fact, *(i) it is almost automatic*; specifically, it requires the user intervention only in few specific cases. *(ii) it is "light"*; specifically, it does not exploit thresholds or weights; as a consequence, it does not need a tuning activity. However, in spite of this "lightness", obtained results are precise and satisfactory, as shown in Section 5. *(iii) it allows the derivation of the various interschema properties within a uniform framework*; such a framework consists of a set of maximum weight matchings computed on suitable bipartite graphs. *(iv) it is specific for XML*; in fact, the framework underlying our approach has been defined for directly covering the XML specificities (see, below, Section 3). *(v) it allows the choice of the "severity level" against which the property extraction task is performed*; such a feature derives from the consideration that applications and scenarios possibly benefiting of derived interschema properties are numerous and extremely various. In some situations the extraction process must be very severe in that it can state the existence of an interschema property between two concepts only if this fact is confirmed by various clues. In other situations, the extraction task can be looser and can assume the existence of an interschema property between two concepts if it has been derived by some computation, without requiring various confirmations. At the beginning of the extraction activity our approach asks the user to specify the desired severity level; this is the only information required to him until the end of the extraction task, when he has to validate obtained results.

It is worth pointing out that, in the past, we have proposed some algorithms for deriving synonymies and homonymies specifically conceived to operate on XML Schemas [5]. They do not exploit thresholds and weights and consider a "severity" level; as a consequence, they follow the same philosophy as the approach we are presenting here; however, they are not able to derive hyponymies and overlappings. In this context the approach presented here can be considered an advancement of this research line and provides a further component allowing the construction of a framework for uniformly deriving a large variety of interschema properties among a great number of XML Schemas.

2 Background

2.1 An overview of the interschema property extraction approaches

In [9] the system CGLUE is proposed. It exploits machine learning techniques for deriving *semantic matchings* between two given ontologies O_1 and O_2. In [11] the authors propose a formal method, based on fuzzy relations, capable of performing the semantic reconciliation of heterogeneous data sources.

In [17] the authors propose Cupid, a system that detects semantic matchings holding between two schemas. First, Cupid represents input schemas by means of trees. Then it computes a coefficient, named *linguistic similarity* for each pair of schema elements. After this Cupid computes the *structural similarity* coefficient by means of a suitable tree-based algorithm. Finally it combines linguistic and structural similarity coefficients to derive semantic matchings.

In [1] the authors describe MOMIS, a system devoted to handle both integration and querying activities on heterogeneous data sources. MOMIS follows a "semantic approach" to interschema property extraction, based on an intensional study of information sources.

In [14] a statistical framework for performing schema matching tasks on Web query interfaces (i.e., on data sources containing the results of the execution of queries posed through Web interfaces) is proposed. In their approach, the authors hypothesize the presence, for each application context, of a "hidden schema model" which acts as a unified generative model describing how schemas are derived from a finite vocabulary of attributes.

In [4] the authors propose a matching algorithm for measuring the structural similarity between an XML document

D and a DTD T. This algorithm assigns a score (called *similarity measure*) to D, indicating how much D is similar to T. The approach represents both D and T as labelled trees. In [20, 21] the system DIKE is proposed. This system is devoted to extract interschema properties from E/R Schemas. DIKE has been conceived to operate with quite a small number of information sources; as a consequence, it privileges accuracy to computation time. This system exploits a support dictionary containing an initial set of (generally lexical) similarities constructed with the support of human experts during a training phase. The extraction task is graph-based and takes into account the "context" of the concepts into examination; it exploits a large variety of thresholds and weights in order to better adapt itself to the sources which it currently operates on; these thresholds and these weights must be tuned during the training phase.

2.2 Classification criteria

In the literature various classification criteria have been proposed for comparing schema matching approaches (see, for example, [23]). They allow the approaches to be examined from various points of view. Specifically, the criteria appearing particularly interesting in our context are the following:

Schema Types. Some matching algorithms can operate only on a specific kind of data sources (e.g., XML, relational, and so on); these approaches are called *specific* in the following. On the contrary, other approaches are able to manage every kind of data sources; we call these approaches *generic* in the following. A generic approach is usually more versatile than a specific one because it can be applied on data sources characterized by heterogeneous representation formats. On the contrary, a specific approach can take advantage of the peculiarities of the corresponding data model.

Instance-Based versus Schema-Based. In order to detect interschema properties, schema matching approaches can consider data instances (i.e., the so-called *extensional information*) or schema-level information (i.e., the so-called *intensional information*). The former class of approaches is called instance-based; the latter one is known as schema-based. An intermediate category is represented by *mixed approaches*, i.e., those ones exploiting both intensional and extensional information. Instance-based approaches are generally very precise because they look at the actual content of the involved sources; however, they are quite expensive since they must examine the extensional component of the involved sources. On the contrary, schema-based approaches look at the intensional information only and, consequently, they are less expensive; however, they could be less precise. Finally, the results of an instance-based approach are valid only for the sources it has been applied to, whereas the results of a schema-based approach are valid for all those sources conforming to the considered schemas. As a consequence, instance-based and mixed approaches are more suited for those application contexts characterized by few sources and requiring very accurate results, whereas schema-based approaches are more suited for those application contexts involving a great number of sources.

Individual versus Combinatorial. An *individual* matcher exploits just one matching criterion; on the contrary, *combinatorial* approaches integrate different individual matchers to perform schema matching activities. Combinatorial matchers can be further classified as: *(i) hybrid matchers*, if they directly combine several schema matching approaches into a unique matcher; *(ii) composite matchers*, if they combine the results of several independently executed matchers; they are sometimes called multi-strategy approaches. The individual matchers are simpler and, consequently, less time consuming than the combinatorial ones; however, the results they obtain are generally less accurate than those returned by combinatorial matchers.

Matching Cardinality. Some approaches have been conceived to derive only semantic similarities between two single components of different schemas (*1:1 matchings*). Other approaches are capable of deriving also semantic similarities between one single component of a schema and a group of components of the other schemas (*1:n matchings*) or between two groups of components of different schemas (*m:n matchings*).

Exploitation of Auxiliary Information. Some approaches could exploit auxiliary information (e.g., dictionaries, thesauruses, and so on) for their activity; on the contrary, this information is not needed in other approaches. Auxiliary information represents an effective way to enrich the knowledge that an approach can exploit. However, in order to maintain its effectiveness, the time required to compile and/or retrieve it must be negligible w.r.t. the time required by the whole approach to perform its matches. For this reason, pre-built or automatically computed auxiliary information would be preferred to the manually provided one.

3 Preliminaries

3.1 Neighborhood definition

We start to illustrate the definition of the neighborhood of an element or an attribute in an XML Schema by introducing the concept of x-component, that allows both elements and attributes of an XML Schema to be uniformly handled.

Definition 3.1 Let S be an XML Schema; an *x-component* of S is either an element or an attribute of S. □

An x-component is characterized by its name, its typology (indicating if it is either a complex element or a simple element or an attribute) and its data type.

Definition 3.2 Let S be an XML Schema; the sets of its x-components, its keys and its keyrefs are denoted as $XCompSet(S)$, $KeySet(S)$ and $KeyrefSet(S)$, respectively. The union of $XCompSet(S)$, $KeySet(S)$ and $KeyrefSet(S)$ is denoted as $ConstructSet(S)$; it forms *the set of constructs* of S. □

We now introduce some functions that allow the strength of the relationship existing between two x-components x_S and x_T of an XML Schema S to be determined. These functions are:

- $veryclose(x_S, x_T)$, that returns *true* if and only if: *(i)* $x_T = x_S$, or *(ii)* x_T is an attribute of x_S, or *(iii)* x_T is a simple sub-element of x_S; in all the other cases it returns *false*;
- $close(x_S, x_T)$, that returns *true* if and only if: *(i)* x_T is a complex sub-element of x_S, or *(ii)* x_S and x_T are two complex elements of S and there exists a `keyref` element stating that an attribute of x_S refers to a `key` attribute of x_T; in all the other cases it returns *false*;
- $near(x_S, x_T)$, that returns *true* if and only if either $veryclose(x_S, x_T) = true$ or $close(x_S, x_T) = true$; in all the other cases it returns *false*;
- $reachable(x_S, x_T)$, that returns *true* if and only if there exists a sequence of *x-components* $x_1, x_2, \ldots, x_n$ such that $x_S = x_1, near(x_1, x_2) = near(x_2, x_3) = \ldots = near(x_{n-1}, x_n) = true, x_n = x_T$; in all the other cases it returns *false*.

We are now able to introduce the concept of Connection Cost from an x-component x_S to an x-component x_T. It is a measure of the correlation degree existing between x_S and x_T and indicates how much the concept expressed by x_T is "close" to the concept represented by x_S.

Definition 3.3 Let S be an XML Schema and let x_S and x_T be two x-components of S. The *Connection Cost* from x_S to x_T, denoted by $CC(x_S, x_T)$, is defined as: *(i)* 0 if $veryclose(x_S, x_T) = true$; *(ii)* 1 if $close(x_S, x_T) = true$; *(iii)* $\mathcal{C}_{\mathcal{ST}}$ if $reachable(x_S, x_T) = true$ and $near(x_S, x_T) = false$; *(iv)* ∞ if $reachable(x_S, x_T) = false$.

Here $\mathcal{C}_{ST} = min_{x_A}\ (CC(x_S, x_A) + CC(x_A, x_T))$ for each x_A such that $reachable(x_S, x_A)$ = $reachable(x_A, x_T) = true$. □

We are now provided with all tools necessary to define the concept of neighborhood of an x-component.

Definition 3.4 Let S be an XML Schema, let x_S be an x-component of S and let j be a non-negative integer. The j^{th} neighborhood of x_S is defined as: $nbh(x_S, j) = \{x_T |\ x_T \in XCompSet(S), CC(x_S, x_T) \leq j\}$ □

The next proposition provides an estimation of the maximum number of distinct neighborhoods for an x-component; the interested reader can find its proof in the Appendix available at the address `http://www.mat.unical.it/terracina/informatica07/Appendix.pdf`.

Proposition 3.1 Let S be an XML Schema; let x_S be an x-component of S; let m be the number of complex elements of S; then $nbh(x_S, j) = nbh(x_S, m-1)$ for each j such that $j \geq m$. □

The next proposition determines the worst case time complexity for computing all neighborhoods of all x-components of an XML Schema S. The interested reader can find its proof in the Appendix.

Proposition 3.2 Let S be an XML Schema and let n be the number of its x-components. The worst case time complexity for computing all neighborhoods of all x-components of S is $O(n^3)$. □

3.2 Neighborhood comparison

Given two x-components x_{1_j} and x_{2_k} and two corresponding neighborhoods $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$, there could exist different relationships between them.

Specifically, three possible relationships, namely *similarity*, *comparability* and *generalization*, could be taken into account. All of them are derived by computing suitable objective functions on the maximum weight matching associated with a bipartite graph obtained from the x-components of $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$.

In the following we indicate by $BG(x_{1_j}, x_{2_k}, v) = \langle NSet(x_{1_j}, x_{2_k}, v), ESet(x_{1_j}, x_{2_k}, v)\rangle$ the bipartite graph associated with $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$; when it is not confusing, we shall use the notation $BG(v)$ instead of $BG(x_{1_j}, x_{2_k}, v)$. In $BG(v)$, $NSet(v) = PSet(v) \cup QSet(v)$ represents the set of nodes; there is a node in $PSet(v)$ (resp., $QSet(v)$) for each x-component of $nbh(x_{1_j}, v)$ (resp., $nbh(x_{2_k}, v)$). $ESet(v)$ is the set of edges; there is an edge between $p \in PSet(v)$ and $q \in QSet(v)$ if: *(i)* a synonymy between the names of the x-components x_p and x_q, associated with p and q, is stored in the reference thesaurus; *(ii)* the cardinalities of x_p and x_q are *compatible*; *(iii)* their data types are *compatible* (this last condition must be verified only if x_p and x_q are attributes or simple elements).

Here, the cardinalities of two x-components are considered compatible if the intersection of the intervals they represent is not empty. The motivation underlying this assumption is that cardinalities represent constraints associated with the involved concepts and, therefore, contribute to define their semantics; as a consequence, completely disjoint intervals are a symptom that the two concepts have different semantics. Compatibility rules associated with data types are analogous to the corresponding ones valid for high level programming languages.

The maximum weight matching for $BG(v)$ is the set $ESet'(v) \subseteq ESet(v)$ of edges such that, for each node $x \in PSet(v) \cup QSet(v)$, there is *at most one* edge of $ESet'(v)$ incident onto x and $|ESet'(v)|$ is maximum (for algorithms solving the maximum weight matching problem, see [12]).

As previously pointed out, in our approach, all neighborhood comparisons are performed by computing the maximum weight matching on a suitable bipartite graph. The reasoning underlying this choice is the following: all types of neighborhood comparison (i.e., similarity, comparability and generalization) aim to determine how much two neighborhoods are someway close. A neighborhood is a set of x-components. Generally speaking, two sets are close if they share a sufficiently large number of their elements. In our application context, two x-components belonging to two different neighborhoods can be considered as a shared x-component if a synonymy exists between them. Then, the maximum weight matching on the bipartite graph constructed from two neighborhoods allows the maximum number of pairs of synonymous x-components belonging to the neighborhoods to be determined; as a consequence, it allows the derivation of the maximum number of x-components that can be considered shared between the two neighborhoods and, therefore, the computation of the closeness degree of the two neighborhoods at hand.

3.2.1 Neighborhood similarity

Intuitively, two neighborhoods (and, more in general, two sets of objects) are considered similar if most of their components are similar.

In order to determine if $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are similar, we construct $BG(x_{1_j}, x_{2_k}, v)$ and, then, compute the objective function $\phi_{BG}(v) = \frac{2|ESet'(v)|}{|PSet(v)|+|QSet(v)|}$.

Here $|ESet'(v)|$ represents the number of matches associated with $BG(v)$, as well as the number of pairs of x-components $\langle x^p_{1_j}, x^q_{2_k}\rangle$ such that $x^p_{1_j} \in nbh(x_{1_j}, v)$, $x^q_{2_k} \in nbh(x_{2_k}, v)$ and a synonymy between the names of $x^p_{1_j}$ and $x^q_{2_k}$ is stored in the reference thesaurus. $|PSet(v)| + |QSet(v)|$ denotes the total number of nodes in $BG(v)$, as well as the total number of x-components associated with $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$. The coefficient 2 at the numerator of ϕ_{BG} is necessary to make the numerator and the denominator comparable; in fact, $|PSet(v)| + |QSet(v)|$ refers to x-components whereas $|ESet'(v)|$ regards pairs of x-components. Finally, $\phi_{BG}(v)$ represents the share of matching nodes in $BG(v)$, as well as the share of similar x-components present in $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$. The formal definition of the neighborhood similarity is given below.

Definition 3.5 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). Two neighborhoods $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are similar if, given the bipartite graph $BG(x_{1_j}, x_{2_k}, v)$, $\phi_{BG}(v) > \frac{1}{2}$. □

This definition assumes that $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are similar if $\phi_{BG}(v) > \frac{1}{2}$; such an assumption derives from the consideration that two sets of objects can be considered similar if the number of similar components is greater than the number of the dissimilar ones or, in other words, if the number of similar components is greater than half of the total number of components.

The following theorem states the worst case time complexity for determining if two neighborhoods are similar. Its proof is provided in the Appendix.

Theorem 3.1 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). Let p be the maximum between $|nbh(x_{1_j}, v)|$ and $|nbh(x_{2_k}, v)|$. The worst case time complexity for determining if $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are similar is $O(p^3)$. □

3.2.2 Neighborhood comparability

Intuitively, two neighborhoods $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are comparable if there exist at least two (quite large) subsets $XSet_j$ of $nbh(x_{1_j}, v)$ and $XSet_k$ of $nbh(x_{2_k}, v)$ that are similar. Similarity between $XSet_j$ and $XSet_k$ is computed by constructing a bipartite graph $\overline{BG}(XSet_j, XSet_k)$ starting from the x-components of $XSet_j$ and $XSet_k$, and by computing $\phi_{\overline{BG}}$ in a way analogous to that we have seen in Section 3.2.1. Comparability is a weaker property than similarity. As a matter of fact, if two neighborhoods are similar, they are also comparable. However, it may be that two neighborhoods are not similar but are comparable because they have quite large similar subsets. The formal definition of neighborhood comparability is provided below.

Definition 3.6 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). Two neighborhoods $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are comparable if there exist two subsets, $XSet_j$ of $nbh(x_{1_j}, v)$ and $XSet_k$ of $nbh(x_{2_k}, v)$, such that: *(i)* $|XSet_j| > \frac{1}{2}|nbh(x_{1_j}, v)|$; *(ii)* $|XSet_k| > \frac{1}{2}|nbh(x_{2_k}, v)|$; *(iii)* $\phi_{\overline{BG}}(XSet_j, XSet_k) > \frac{1}{2}$. □

In this definition, conditions *(i)* and *(ii)* guarantee that $XSet_j$ and $XSet_k$ are representative (i.e., quite large); we assume that this happens if they involve more than half of the components of the corresponding neighborhoods. Finally, condition *(iii)* guarantees that $XSet_j$ and $XSet_k$ are similar.

The following theorem states the worst case time complexity for verifying if two neighborhoods are comparable. Its proof can be found in the Appendix.

Theorem 3.2 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). Let p be the maximum between $|nbh(x_{1_j}, v)|$ and $|nbh(x_{2_k}, v)|$. The worst case time complexity for determining if $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are comparable is $O(p^3)$. □

Corollary 3.1 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). If $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are similar, then they are also comparable. □

3.2.3 Neighborhood generalization

Consider two neighborhoods α and β and assume that: *(1)* they are not similar; *(2)* most of the x-components of β match with x-components of α; *(3)* most of the x-components of α do not match with x-components of β. If all these conditions hold, then it is possible to conclude that the reality represented by α is richer than that represented by β and, consequently, that α is more specific than β or, conversely, that β is more general than α. As an example, α could be the set of attributes and sub-elements describing the concept *PhD Student* whereas β might be the set of attributes and sub-elements describing the concept *Student*. The following definition formalizes this reasoning.

Definition 3.7 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). We say that $nbh(x_{1_j}, v)$ is more specific than $nbh(x_{2_k}, v)$ (and, consequently, that $nbh(x_{2_k}, v)$ is more general than $nbh(x_{1_j}, v)$) if: *(i)* they are not similar and *(ii)* the objective function $\varphi_{BG}(x_{1_j}, x_{2_k}, v) = \frac{|ESet'(v)|}{|QSet(v)|}$, associated with the bipartite graph $BG(x_{1_j}, x_{2_k}, v)$, is greater than $\frac{1}{2}$; here $BG(x_{1_j}, x_{2_k}, v)$ has been described in Section 3.2, $ESet'(v)$ represents the set of matching edges associated with BG whereas $QSet(v)$ is the set of nodes of BG corresponding to the x-components of $nbh(x_{2_k}, v)$. □

The reasoning underlying Definition 3.7 derives from the observation that $\varphi_{BG}(x_{1_j}, x_{2_k}, v)$ represents the share of x-components belonging to $nbh(x_{2_k}, v)$ matching with the x-components of $nbh(x_{1_j}, v)$. If this share is sufficiently high then most of the x-components of $nbh(x_{2_k}, v)$ match with the x-components of $nbh(x_{1_j}, v)$ (condition *(2)*) but, since $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are not similar (condition *(1)*), most of the x-components of $nbh(x_{1_j}, v)$ do not match with the x-components of $nbh(x_{2_k}, v)$ (condition *(3)*). As a consequence, it is possible to conclude that $nbh(x_{1_j}, v)$ is more specific than $nbh(x_{2_k}, v)$ or, conversely, that $nbh(x_{2_k})$ is more general than $nbh(x_{1_j}, v)$.

The following theorem states the worst case time complexity for verifying if a neighborhood is more specific than another one. Its proof is provided in the Appendix.

Theorem 3.3 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). Let p be the maximum between $|nbh(x_{1_j}, v)|$ and $|nbh(x_{2_k}, v)|$. The worst case time complexity for determining if $nbh(x_{1_j}, v)$ is more specific than $nbh(x_{2_k}, v)$ is $O(p^3)$. □

4 Extraction of interschema properties

In this section we illustrate our approach for the extraction of interschema properties. As pointed out in the Introduction, we shall concentrate our attention on the following properties: *(i) Synonymies*: a synonymy indicates that two x-components have the same meaning. *(ii) Hyponymies/Hypernymies*: given two x-components x_S and x_T, x_S is a hyponym of x_T (that is, in its turn, the hypernym of x_S) if x_S has a more specific meaning than x_T. *(iii) Overlappings*: roughly speaking, given two x-components x_S and x_T, an overlapping holds between them if they are neither synonymous nor one a hyponym of the other but there exist non-empty sets of attributes and sub-elements $\{x_{S_1}, x_{S_2}, \ldots, x_{S_n}\}$ of x_S and $\{x_{T_1}, x_{T_2}, \ldots, x_{T_n}\}$ of x_T such that, for $1 \leq i \leq n$, x_{S_i} is synonymous with x_{T_i}. *(iv) Homonymies*: an homonymy states that two x-components have the same name and the same typology, but different meanings.

Our approach exploits a thesaurus storing lexical synonymies holding among the terms of a language; specifically, it uses the English language and WordNet [19]. If necessary, different (possibly existing) domain-specific thesauruses could be used in the prototype implementing our approach; they can be provided by means of a suitable, friendly interface.

4.1 Derivation of candidate pairs

In order to verify if an interschema property holds between two x-components x_{1_j}, belonging to S_1, and x_{2_k}, belonging to S_2, it is necessary to examine their neighborhoods. Specifically, our approach operates as follows. First it considers $nbh(x_{1_j}, 0)$ and $nbh(x_{2_k}, 0)$ and verifies if they are comparable. In the affirmative case, it is possible to conclude that x_{1_j} and x_{2_k} refer to analogous "contexts" and, presumably, define comparable concepts. As a consequence, the pair $\langle x_{1_j}, x_{2_k} \rangle$ is marked as *candidate* for an interschema property. However, observe that $nbh(x_{1_j}, 0)$ (resp., $nbh(x_{2_k}, 0)$) takes only attributes and simple sub-elements of x_{1_j} (resp., x_{2_k}) into account; as a consequence, it considers quite a limited context. If a higher severity level is required, it is necessary to verify that other neighborhoods of x_{1_j} and x_{2_k} are comparable before marking the pair $\langle x_{1_j}, x_{2_k} \rangle$ as candidate. Such a reasoning is formalized by the following definition.

Definition 4.1 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). Let u be a severity level. We say that *the pair* $\langle x_{1_j}, x_{2_k} \rangle$ *is candidate for an interschema property* at the severity level u if $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are comparable for each v such that $0 \leq v \leq u$. □

It is possible to define a boolean function $candidate$ that receives two x-components x_{1_j} and x_{2_k} and an integer u and returns *true* if $\langle x_{1_j}, x_{2_k} \rangle$ is a candidate pair at the severity level u, *false* otherwise.

The following theorem states the computational complexity for the detection of candidate pairs. Its proof is immediate from Theorem 3.2 and Definition 4.1.

Theorem 4.1 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). Let u

be a severity level. Finally, let p be the maximum between $|nbh(x_{1_j}, u)|$ and $|nbh(x_{2_k}, u)|$. The worst case time complexity for verifying if $\langle x_{1_j}, x_{2_k} \rangle$ is a candidate pair at the severity level u is $O(u \times p^3)$. □

4.2 Derivation of synonymies, hyponymies, overlappings and homonymies

Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). In order to verify if a synonymy, a hyponymy, an overlapping or an homonymy holds between x_{1_j} and x_{2_k} it is necessary to examine their neighborhoods and to determine the relationships holding among them. The following definition formalizes this reasoning:

Definition 4.2 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2) and let u be a severity level.

- A synonymy holds between x_{1_j} and x_{2_k} at the severity level u if: *(i)* $candidate(x_{1_j}, x_{2_k}, u) = true$; *(ii)* $nbh(x_{1_j}, v)$ and $nbh(x_{2_k}, v)$ are similar for each v such that $0 \leq v \leq u$ (see Section 3.2.1).
- x_{1_j} is said a hyponym of x_{2_k} (that, in its turn, is said a hypernym of x_{1_j}) at the severity level u if: *(i)* $candidate(x_{1_j}, x_{2_k}, u) = true$; *(ii)* $nbh(x_{1_j}, 0)$ is more specific than $nbh(x_{2_k}, 0)$ (see Section 3.2.3).
- An overlapping holds between x_{1_j} and x_{2_k} at the severity level u if: *(i)* $candidate(x_{1_j}, x_{2_k}, u) = true$; *(ii)* x_{1_j} and x_{2_k} are not synonymous; *(iii)* x_{1_j} is neither a hyponym nor a hypernym of x_{2_k}.
- An homonymy holds between x_{1_j} and x_{2_k} at the severity level u if: *(i)* $candidate(x_{1_j}, x_{2_k}, u) = false$; *(ii)* x_{1_j} and x_{2_k} have the same name; *(iii)* x_{1_j} and x_{2_k} are both elements or both attributes. □

It is possible to define a boolean function $synonymy(x_{1_j}, x_{2_k}, u)$ (resp., $hyponymy(x_{1_j}, x_{2_k}, u)$, $overlapping(x_{1_j}, x_{2_k}, u)$, $homonymy(x_{1_j}, x_{2_k}, u)$), that receives two x-components x_{1_j} and x_{2_k} and an integer u and returns *true* if a synonymy (resp., a hyponymy, an overlapping, an homonymy) holds between x_{1_j} and x_{2_k} at the severity level u, *false* otherwise.

As for the computational complexity of the interschema property derivation, it is possible to state the following theorem whose proof can be found in the Appendix.

Theorem 4.2 Let S_1 and S_2 be two XML Schemas. Let x_{1_j} (resp., x_{2_k}) be an x-component of S_1 (resp., S_2). Let u be a severity level. Finally, let p be the maximum between $|nbh(x_{1_j}, u)|$ and $|nbh(x_{2_k}, u)|$. The worst case time complexity for computing $synonymy(x_{1_j}, x_{2_k}, u)$, $hyponymy(x_{1_j}, x_{2_k}, u)$, $overlapping(x_{1_j}, x_{2_k}, u)$, $homonymy(x_{1_j}, x_{2_k}, u)$ is $O(u \times p^3)$. □

Corollary 4.1 Let S_1 and S_2 be two XML Schemas. Let u be a severity level. Let m be the maximum between the number of complex elements of S_1 and S_2. Finally, let q be the maximum cardinality of a neighborhood of S_1 or S_2. The worst case time complexity for deriving all interschema properties holding between S_1 and S_2 at the severity level u is $O(u \times q^3 \times m^2)$. □

5 Experimental results

5.1 Introduction

In this section we provide a detailed description of the experiments we have carried out for testing the performance of our approach. We have performed a large variety of experiments, devoted to test the various aspects of our approach; they will be presented in the next subsections.

It is worth pointing out that some of our tests have been inspired to ideas and methodologies illustrated in [6]; in this paper, the authors propose a catalogue of criteria for comparing some of the most popular interschema property extraction systems, namely, *Autoplex* [2], *Automatch* [3], *COMA* [7], *Cupid* [17], LSD [8], GLUE [10], *SemInt* [16] and SF (*Similarity Flooding*) [18]. In our opinion, this is a very interesting effort and we have decided to exploit the same criteria (and, whenever possible, the same sources) for testing the performance of our approach. This choice allowed us to obtain an objective evaluation of our approach, as well as to make a precise comparison between it and the other systems evaluated by [6].

5.2 Characteristics of the exploited sources

In our tests we have exploited a large variety of XML Schemas relating to disparate application contexts; specifically, we have considered XML Schemas relating to Biomedical Data, Project Management, Property Register, Industrial Companies, Universities, Airlines, Scientific Publications and Biological Data. In our tests, we have compared all pairs of XML schemas within a particular domain.

Biomedical Schemas have been derived from various sites; among them we cite: `http://www.biomediator.org`. XML Schemas relating to Project Management, Property Register and Industrial Companies have been derived from Italian Central Government Office sources and are shown at the address: `http://www.mat.unical.it/terracina/tests.html`. XML Schemas relating to Universities have been downloaded from the address: `http://anhai.cs.uiuc.edu/archive/domains/courses.html`. XML Schemas relating to Airlines have been found in [22]. XML Schemas relating to Scientific Publications have been supplied by the authors of [15]. Finally, Biological Schemas have been downloaded from the addresses: `http://smi-web.stanford.edu/projects/`

helix/pubs/ismb02/schemas/, http://www.cs.toronto.edu/db/clio/data/GeneX_RDB-s.xsd and http://www.genome.ad.jp/kegg/soap/v3.0/KEGG.wsdl.

As far as exploited thesauruses are concerned, we have used WordNet for XML Schemas relating to Project Management, Property Register, Industrial Companies, Universities, Airlines and Scientific Publications. On the contrary, for Biomedical and Biological Schemas we have exploited the Biocomplexity Thesaurus, a biological domain specific thesaurus available at the address: http://thesaurus.nbii.gov.

Examined sources were characterized by the following properties, expressed according to the terminology and the measures of [6]:

Number of schemas: we have considered 35 XML Schemas whose characteristics are reported in Table 1; this number of schemas is quite similar to those considered by the authors of the other approaches for performing their evaluation; they are reported in Table 2. From this table it is possible to see that the number of schemas exploited by the other approaches for carrying out their evaluation activity ranges from 2 to 24.

Size of schemas: the size of the evaluated XML Schemas, i.e., the number of their elements and attributes, ranges from 12 to 645. The minimum, the maximum and the average size of the sources exploited for evaluating the other approaches, derived by [6], are reported in Table 2[1]. An analysis of this table shows that the sizes of the schemas evaluated by our approach are quite close to those of the sources examined by the other systems. The size of a test schema is relevant because it influences the quality of obtained results; in fact, as mentioned in [6], the bigger the input schemas are, the greater the search space for candidate pairs is and the lower the quality of obtained results will be.

The number of comparisons we have carried out for each domain are shown in the last column of Table 1.

5.3 Accuracy Measures exploited in our experimental tests

All accuracy measures proposed in [6] and computed during our experiments have been obtained according to the following general framework: *(i)* a set of experts has been asked to identify interschema properties existing among involved XML Schemas; *(ii)* interschema properties among the same XML Schemas have been determined by the approach to evaluate; *(iii)* the properties provided by the experts and those returned by the approach to test have been compared and accuracy measures have been computed.

The number of experts that have been involved in manually solving the match tasks is as follows: 6 for Biomedical Data, 3 for Project Management, 3 for Property Register, 4 for Industrial Companies, 4 for Universities, 2 for Airlines, 2 for Scientific Publications and 7 for Biological Data.

Let A be the set of properties provided by the experts and let C be the set of properties returned by the approach to evaluate; two basic accuracy measures are: *(i) Precision* (hereafter *Pre*), that specifies the share of correct properties detected by the system among those it derived. It is defined as: $Pre = \frac{|A \cap C|}{|C|}$. *(ii) Recall* (hereafter *Rec*), that indicates the share of correct properties detected by the system among those the experts provided. It is defined as: $Rec = \frac{|A \cap C|}{|A|}$. Precision and Recall are typical measures of Information Retrieval (see [25]). Both of them fall within the interval $[0, 1]$; in the ideal case (i.e., when $A \equiv C$) they are both equal to 1. It is worth noting that the set C of interschema properties our approach derives varies with the severity level; in order to make this evident, we shall use the symbol $C(u)$ instead of C. However, as pointed out in [6], neither Precision nor Recall alone can accurately measure the quality of an interschema property extraction algorithm; in order to improve the result accuracy, it appears necessary to consider a joint measure of them. Two very popular measures satisfying these requirements are: *(i) F-Measure* [3, 25], that represents the harmonic mean between Precision and Recall. It is defined as: $F\text{-}Measure = 2 \cdot \frac{Pre \cdot Rec}{Pre + Rec}$. *(ii) Overall* [6, 18], that measures the post-match effort needed for adding false negatives and removing false positives from the set of properties returned by the system to evaluate. It is defined as: $Overall = Rec \cdot (2 - \frac{1}{Pre})$. F-Measure falls within the interval $[0, 1]$ whereas Overall ranges between $-\infty$ and 1; the higher F-Measure (resp., Overall) is, the better the accuracy of the tested approach will be.

5.4 Discussion of obtained results

As for the evaluation of Precision and Recall associated with our approach, we argued that, due to its philosophy and intrinsic structure, an increase of the severity level should have caused an increase of its Precision and a decrease of its Recall. This intuition is motivated by considering that $C(u + 1) \subseteq C(u)$ and that $C(u + 1)$ is obtained from $C(u)$ by eliminating the weakest properties; this should cause $C(u + 1)$ to be more precise than $C(u)$. However, this filtering task could erroneously discard some valid properties; for this reason $C(u + 1)$ could have a smaller Recall than $C(u)$.

In order to verify this intuition and, possibly, to quantify it, we have applied our approach on our test Schemas and we have computed the Average Precision, the Average Recall, the Average F-Measure and the Average Overall at various severity levels. Obtained results are shown in Table 3. From the analysis of this table we can draw the following conclusions:

– As for the severity level 0, *(i)* Precision shows its lowest value; as a consequence, our approach returns some false positives; *(ii)* Recall assumes its highest

[1] The size of a relational source has been intended as the number of its relations and attributes.

Application context	*Number of Schemas*	*Max Depth*	*Minimum, Average and Maximum Number of x-components*	*Minimum, Average and Maximum Number of complex elements*	*Total Number of Comparisons*
Biomedical Data	11	8	15 - 26 - 38	4 - 8 - 16	55
Project Management	3	4	37 - 40 - 42	6 - 7 - 8	3
Property Register	2	4	64 - 70 - 75	14 - 14 - 14	1
Industrial Companies	5	4	23 - 28 - 46	6 - 8 - 9	10
Universities	5	5	15 - 17 - 19	3 - 4 - 5	10
Airlines	2	4	12 - 13 - 13	4 - 4 - 4	1
Scientific Publications	2	6	17 - 18 - 18	8 - 9 - 9	1
Biological Data	5	8	250 - 327 - 645	36 - 60 - 206	10

Table 1: Characteristics of the XML Schemas exploited for testing the performance of our approach

System	*Typology of tested Schema*	*Number of Schemas*	*Minimum size of Schemas*	*Maximum size of Schemas*	*Average size of Schemas*
Our system	XML	35	12	645	70
Autoplex & Automatch	Relational	15	-	-	-
COMA	XML	5	40	145	77
Cupid	XML	2	40	54	47
LSD	XML	24	14	66	-
GLUE	XML	3	34	333	143
SemInt	Relational	10	6	260	57
SF	XML	18	5	22	12

Table 2: Characteristics of the XML Schemas exploited by the other approaches for their evaluation activity

Property Typology	*Average Precision*	*Average Recall*	*Average F-Measure*	*Average Overall*
Severity Level 0	0.86	0.97	0.91	0.81
Severity Level 1	0.96	0.81	0.88	0.78
Severity Level 2	0.97	0.77	0.86	0.75
Severity Level 3	0.97	0.72	0.83	0.70

Table 3: Accuracy measures associated with our approach at various severity levels

value; as a consequence, our approach returns almost all valid properties or, in other words, it returns a very small number of false negatives.

- If the severity level is 1, *(i)* the set of properties returned by our approach contains a smaller number of false positives than the previous case; specifically, it is possible to observe that Precision increases to 0.96; *(ii)* Recall decreases of about 16% w.r.t. the previous case; in other words, a certain increase of false negatives can be observed.

- As for the severity level 2, *(i)* Precision slightly increases to 0.97; *(ii)* Recall decreases of about 5% w.r.t. the previous case.

- When the severity level is equal to 3, *(i)* Precision saturates at its highest value, i.e., 0.97; *(ii)* Recall presents the same trend as the previous case; specifically, a further decrease is observed.

All these experiments confirm our original intuition about the trend of Precision and Recall in presence of variations of the severity level.

From the examination of Table 3 we observe that passing from low to high severity levels causes an increase of the Precision and a corresponding decrease of the Recall. This behaviour is explained by considering that, at low severity levels, a user is willing to accept false positives if this allows him to obtain a complete set of similarities. On the contrary, at high severity levels, a user is willing to receive an incomplete set of similarities by the system but he desires that proposed properties are (almost surely) correct.

Table 3 shows also the great importance of the severity level that provides our approach with a high flexibility. As a matter of fact, in real cases, there are many application contexts where having a high Recall is more important than achieving a high Precision; in these cases our approach can be applied with a severity level equal to 0. On the contrary, there are other situations where obtaining a high Precision is more relevant than having a high Recall; in these situations the user might: *(i)* obtain it automatically by setting an adequate, presumably high, severity level; in this way the automaticity of the approach is preserved but its Recall decreases; *(ii)* obtain it semi-automatically by setting a low severity level and by performing a further, deep, validation of obtained results; in this way the Recall of the approach is preserved but the time the user needs for validation sensibly increases.

After this, we have compared the accuracy of our approach w.r.t. that of the other approaches evaluated in [6]; obtained results are reported in Table 4. We point out that the accuracy measure of the other systems shown in that table have been directly derived from [6]. The only missing data regard Cupid; in fact, in [6], the Authors provide only a qualitative analysis of this system without specifying any quantitative value of its Precision, its Recall, its F-Measure and its Overall. However, a quantitative analysis of Cupid can be found in [13]; in that paper the Authors claim that, for the schemas considered by them, Cupid showed a Precision equal to 0.60, a Recall equal to 0.55, an F-Measure equal to 0.57 and an Overall equal to 0.18; in order to allow a more precise comparison, we must say that the Authors of [13] applied also COMA on the same sources considered for Cupid and, for these sources, they obtained a Precision equal to 0.82, a Recall equal to 0.75, an F-Measure equal to 0.78 and an Overall equal to 0.59.

From the analysis of Table 4 we can observe that: *(i)* at the severity level 0 the Precision of our approach is satisfactory, even if COMA presents a better value; at the severity level 1 our approach has the highest Precision; *(ii)* at the severity level 0 our approach shows the best Recall; on the contrary, at the severity level 1, the Recall of our approach significantly decreases; *(iii)* at the severity level 0 our approach presents, along with COMA, the highest values of F-Measure and Overall; both these two accuracy measures slightly decrease at the severity level 1.

As a conclusion, in our opinion, all these experiments agree on determining that the accuracy of our approach is extremely satisfactory and promising. In addition, our approach shows a great flexibility in that it can be adapted for obtaining the best Precision or the best Recall, according to the exigencies of the application context it is operating in. These results are even more relevant if we take into account that both measures and most of the test sources we have considered had been already uniformly exploited for evaluating a large variety of existing approaches.

We have also verified if the accuracy of our approach depends on the application domain which the test Schemas belong to. The results we have obtained are shown in Figure 1. From the analysis of this figure, it is possible to conclude that the accuracy of our approach is substantially independent of the application domain. As far as our experiments are concerned, we have obtained the best accuracy for the Property Register domain; here, Precision reaches its best value at the severity level 3 and is 0.99; Recall, F-Measure and Overall are maximum at the severity level 0 and are 0.99, 0.94 and 0.87, respectively. We have obtained the worst accuracy in the Biological domain; here, Precision is maximum at the severity level 3 and is 0.87; Recall, F-Measure and Overall reach their best values at the severity level 0 and are 0.87, 0.82 and 0.62, respectively.

5.5 Robustness analysis

5.5.1 Robustness against structural dissimilarities

XML is inherently hierarchical; it allows nested, possibly complex, structures to be exploited for representing a domain. As a consequence, two human experts might model the same reality by means of two XML Schemas characterized by deep structural dissimilarities. We have performed a robustness analysis of our approach, devoted to verify if it is resilient to structural dissimilarities. Before describing our experimental tests about this issue, we point out that the specific features of our approach make it intrinsically robust for the following two cases, that are very common in practice:

- If the typology of an x-component x_{1_j} of an XML Schema S_1 changes from "simple element" to "attribute", or vice versa, no modifications of the interschema properties involving x-components of S_1 occur. This result directly derives from the definition of the function $veryclose$ (see Section 3).
- If x_{1_j} and x'_{1_j} are two complex elements of the same XML Schema S_1 such that x'_{1_j} is a sub-element of x_{1_j} and if S_1 is modified in such a way that x'_{1_j} is no longer a sub-element of x_{1_j} but there exists a `keyref` relating x_{1_j} to x'_{1_j}, then no modifications of the interschema properties involving x-components of S_1 occur. An analogous reasoning holds for the opposite change. This result directly derives from the definition of the function $close$ (see Section 3).

There are further structural modifications that could influence the results of our approach and for which it is not intrinsically robust; for these cases an experimental measure of its robustness appears necessary. Two of the most common structural modifications are analyzed in the following.

Flattening of x-components. Consider Figure 2 illustrating two portions of XML Schemas representing persons. Specifically, in the first XML Schema, the concept "Person" is represented by means of a nested hierarchical structure; on the contrary, in the second XML Schema, the same concept is represented by means of a flat structure.

In order to determine the robustness of our approach against errors occurring owing to the flattening of x-components, for each pair of XML Schemas into consideration, we have progressively altered the structure of one of the XML Schemas by transforming a certain percentage of its x-components from a nested structure to a flat one. For each of these transformations, we have derived the interschema properties associated with the "modified" versions of the XML Schemas and we have computed the corresponding values of the accuracy measures. Specifically, we have considered five cases, corresponding to a percentage of flattened x-components (hereafter FXP - *Flattened X-component Percentage*) equal to: *(a)* 0%; *(b)* 7%; *(c)* 14%; *(d)* 21%; *(e)* 28%. The results we have obtained are shown in Figure 3.

From the analysis of this figure it is possible to observe that our approach shows a good robustness against increases of FXP. As a matter of fact, even if structural dissimilarities occur, the changes in the accuracy measures are generally quite small. In fact, the maximum decrement of the Average Precision (resp., Average Recall, Average F-Measure, Average Overall) w.r.t. case *(a)* is equal to 0.11 (resp., 0.16, 0.13, 0.24) and can be found at the severity level 1 (resp., 0, 2, 0). However, we stress that if the increases of FXP would be significantly greater than those considered above, the changes in the accuracy measures could be significant; this behaviour is correct since it guarantees that our approach shows the right degree of sensitivity to changes to the structure of involved XML Schemas.

Exchange of nesting levels. Consider Figure 4 illustrating two portions of XML Schemas representing catalogues. In the first XML Schema, a catalogue is organized by grouping involved models by brands and, then, by product

System	*Precision*	*Recall*	*F-Measure*	*Overall*
Our system (severity level 0)	0.86	0.97	0.91	0.81
Our system (severity level 1)	0.96	0.81	0.88	0.78
Autoplex & Automatch	0.84	0.82	0.82 & 0.72	0.66
COMA	0.93	0.89	0.90	0.82
Cupid	–	–	–	–
LSD	~ 0.80	0.80	~ 0.80	~ 0.60
GLUE	~ 0.80	0.80	~ 0.80	~ 0.60
SemInt	0.78	0.86	0.81	0.48
SF	–	–	–	~ 0.60

Table 4: Comparison of the accuracy of our approach w.r.t. that of the other approaches evaluated in [6]

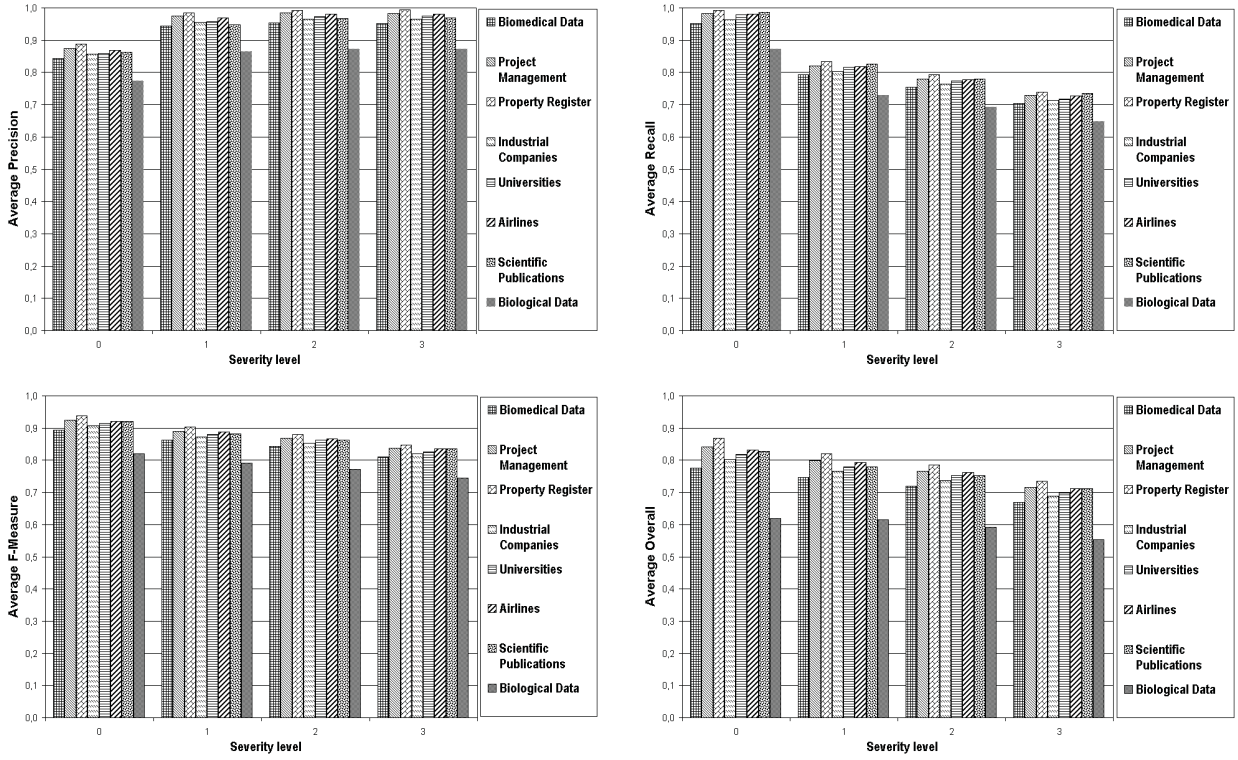

Figure 1: Average Precision, Average Recall, Average F-Measure, Average Overall of our approach in different domains

categories; on the contrary, in the second XML Schema, the same catalogue is organized by grouping involved models by product categories and, then, by brands. In this case we have that two x-components, namely "brand" and "product_category" exchanged their nesting levels within the corresponding XML Schemas. Clearly, an exchange of nesting levels may occur only between complex elements.

Note that the exchange of nesting levels between two complex elements is not always "safe" from a semantical point of view. In fact, consider, again, the first XML Schema of Figure 4, and assume the nesting level of "catalogue" and "brand" to be exchanged; in this case, the semantics of the resulting XML Schema would be quite different w.r.t. that of the original XML Schema; in fact, the new XML Schema would represent a list of brands each of which associated with a separate catalogue of products. Therefore, as for the robustness of our approach in the management of this kind of structural modification, we could expect a decrease of performance w.r.t. the previous case because the semantic modifications produced by the exchange of nesting levels are deeper than those caused by the flattening of x-components.

Our approach is partially intrinsically robust against this kind of structural modification. In fact, our definition of neighborhood, which is the core of our interschema property extraction technique, puts in the same set all the x-components laying at a "distance" *less than or equal to* j from the component under consideration (see Definition 3.4). Now, consider an x-component x_S and assume that an exchange of nesting levels occurs between two of its sub-elements, say x'_S and x''_S, laying at distance j and $j+1$ from x_S, respectively; this structural modification will imply some differences in $nbh(x_S, j)$ but not in $nbh(x_S, j+1)$. Specifically, $nbh(x_S, j)$ contains x'_S before the exchange of nesting levels whereas it contains x''_S after the exchange; by contrast, $nbh(x_S, j+1)$ contains x'_S and x''_S both before and after the exchange. This implies that a possible error in the evaluation of the interschema properties involving x_S may occur only when the j^{th} neighborhood of x_S is considered; however, this error is not propagated through

```
<xs:element name="person">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="address"/>
        </xs:sequence>
        <xs:attribute name="first_name" type="xs:string"/>
        <xs:attribute name="last_name" type="xs:string"/>
        <xs:attribute name="gender" type="xs:string"/>
        <xs:attribute name="birthdate" type="xs:date"/>
    </xs:complexType>
</xs:element>
<xs:element name="address">
    <xs:complexType>
        <xs:attribute name="city" type="xs:string"/>
        <xs:attribute name="state" type="xs:string"/>
        <xs:attribute name="country" type="xs:string"/>
        <xs:attribute name="zip" type="xs:string"/>
    </xs:complexType>
</xs:element>
```

```
<xs:element name="person">
    <xs:complexType>
        <xs:attribute name="first_name" type="xs:string"/>
        <xs:attribute name="last_name" type="xs:string"/>
        <xs:attribute name="gender" type="xs:string"/>
        <xs:attribute name="birthdate" type="xs:date"/>
        <xs:attribute name="city" type="xs:string"/>
        <xs:attribute name="state" type="xs:string"/>
        <xs:attribute name="country" type="xs:string"/>
        <xs:attribute name="zip" type="xs:string"/>
    </xs:complexType>
</xs:element>
```

Figure 2: Example of "nested" and "flat" structures

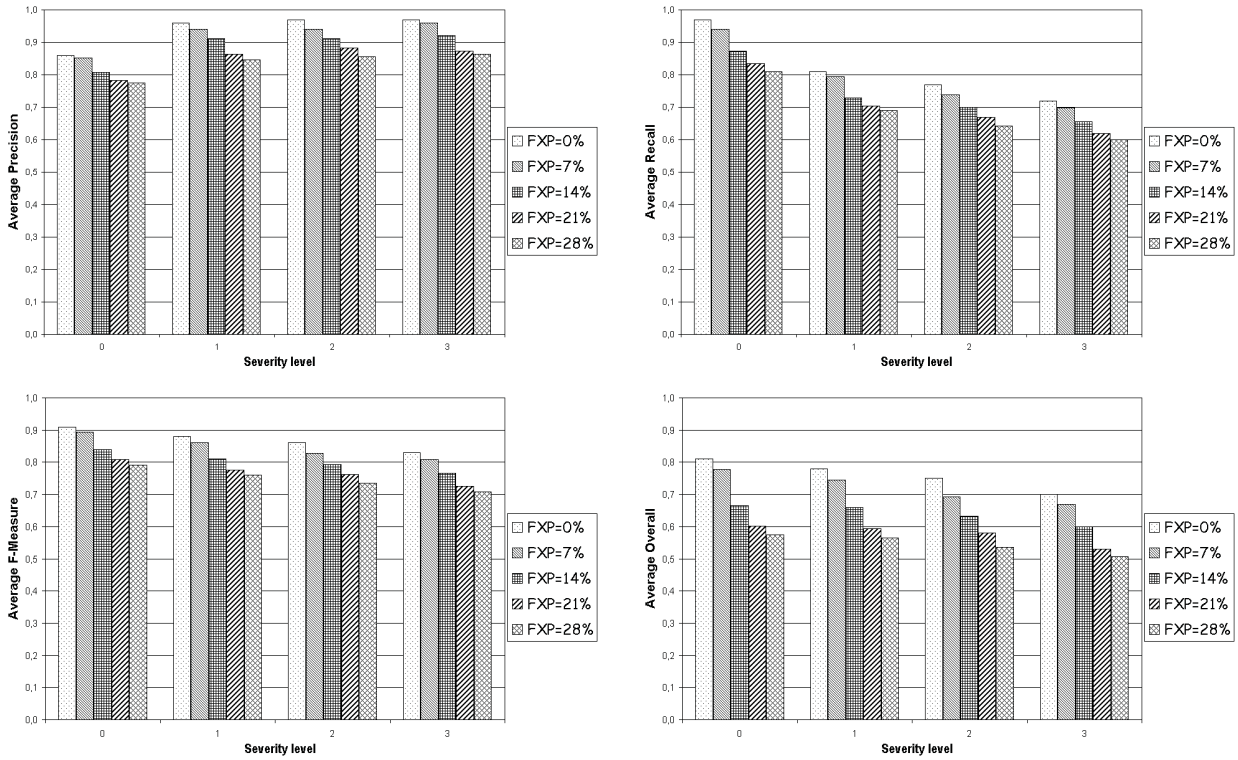

Figure 3: Average Precision, Average Recall, Average F-Measure, Average Overall for various values of FXP

the next neighborhoods. This important feature of our approach mitigates the possible problems arising from the structural modifications caused by the exchange of nesting levels.

In order to quantitatively evaluate the robustness of our approach against errors caused by the exchange of nesting levels, for each pair of XML Schemas into consideration, we have progressively altered the structure of one of the XML Schemas of the pair by exchanging the nesting level of a certain percentage of its x-components. For each of these transformations, we have derived the interschema properties associated with the "modified" versions of the XML Schemas and we have computed the corresponding values of the accuracy measures. Specifically, we have considered five cases, corresponding to a percentage of exchanged nesting levels (hereafter ENP - *Exchanged Nesting level Percentage*) equal to: *(a)* 0%; *(b)* 7%; *(c)* 14%; *(d)* 21%; *(e)* 28%. The results that we have obtained are shown in Figure 5.

From the analysis of this figure it is possible to observe that the robustness of our approach against increases of ENP is satisfactory, even if, as expected, its overall performance is slightly worse than that obtained for the same percentage of FXP. In any case, the changes of the accuracy measures caused by an increase of ENP are generally acceptable. In fact, the maximum decrement of the Average Precision (resp., Average Recall, Average F-Measure, Average Overall) w.r.t. case *(a)* is equal to 0.17 (resp., 0.21, 0.18, 0.33) and can be found at the severity level 2 (resp., 0, 2, 0). However, analogously to the previous case, if the increases of ENP would be quite high, the variations of the semantics of the corresponding XML Schemas would be also significant and, consequently, the accuracy measures might significantly decrease; however, we point out

```
<xs:element name="catalogue">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="brand"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="brand">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="product_category"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="product_category">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="model"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="model">
    <xs:complexType>
        <xs:attribute name="detail" type="xs:string"/>
        <xs:attribute name="price" type="xs:string"/>
    </xs:complexType>
</xs:element>
```

```
<xs:element name="catalogue">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="product_category"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="product_category">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="brand"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="brand">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="model"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="model">
    <xs:complexType>
        <xs:attribute name="detail" type="xs:string"/>
        <xs:attribute name="price" type="xs:string"/>
    </xs:complexType>
</xs:element>
```

Figure 4: Example of exchange of nesting levels

that this behaviour is desirable since it proves, again, that our approach shows a good degree of sensitivity against changes of the structure of involved XML Schemas.

5.5.2 Robustness against thesaurus errors

In this experiment we have tested the effects of errors and inaccuracies in the thesaurus received in input by our approach. Specifically, we have asked experts to validate the similarities contained in the input thesauruses and involving elements and attributes of the considered XML Schemas in such a way to remove any possible error.

After this, we have performed some variations on the corrected thesauruses and, for each of them, we have computed Average Precision, Average Recall, Average F-Measure and Average Overall of our system. Variations we have carried out on the correct thesauruses are: *(a)* 10% of correct similarities have been filtered out; *(b)* 20% of correct similarities have been filtered out; *(c)* 30% of correct similarities have been filtered out; *(d)* 50% of correct similarities have been filtered out; *(e)* 10% of wrong similarities have been added; *(f)* 20% of wrong similarities have been added; *(g)* 30% of wrong similarities have been added; *(h)* 50% of wrong similarities have been added.

Table 5 presents the values of the Average Precision, the Average Recall, the Average F-Measure and the Average Overall we have obtained for the extraction of the interschema properties at various severity levels. These results show that our system is quite robust w.r.t. errors and inaccuracies in the thesauruses provided in input. At the same time it shows a good sensitivity against errors because, if the correct similarities that are filtered out or the wrong similarities that are added are excessive, the system accuracy significantly decreases.

6 Comparison between our approach and the related ones illustrated in Section 2

In this section we compare our approach with the related ones already illustrated in Section 2.

CGLUE. The only similarity between our approach and CGLUE concerns the exploitation of *auxiliary information*; in particular, CGLUE uses the *training matches* (i.e., semantic matches provided by the users for training its learners) whereas our approach exploits a thesaurus. As for differences existing between them, we may observe that: *(i)* CGLUE exploits machine learning techniques, whereas our approach is based on graph matching algorithms; *(ii)* CGLUE is *generic* whereas our approach is specialized for XML sources; *(iii)* CGLUE is both *schema-based* and *instance-based*; as a consequence, it requires a deep analysis of data instances; by contrast, our approach is *schema-based*; *(iv)* CGLUE is *composite* in that it combines various algorithms for detecting semantic matches; by contrast, our approach is *hybrid*; *(v)* CGLUE was conceived for detecting *1:1*, *1:n* and *n:m* matchings, whereas our approach aims to derive *1:1 matchings*.

Approach of [11]. The approach of [11] and ours share some similarities; specifically, *(i)* both of them are *hybrid*; *(ii)* both of them were conceived for detecting *1:1 matchings*. As for differences, we may observe that: *(i)* in order to carry out its tasks, the approach of [11] exploits fuzzy tools whereas our approach uses graph-based techniques; *(ii)* the approach of [11] is *generic*, i.e., it is not specialized for XML sources; *(iii)* the approach of [11] is *instance-based* whereas our approach is *schema-based*; *(iv)* the approach of [11] does not require *auxiliary information*.

Cupid. As for similarities between our approach and Cupid, we may notice that: *(i)* in both of them schema elements are matched in a pair-wise manner by means of

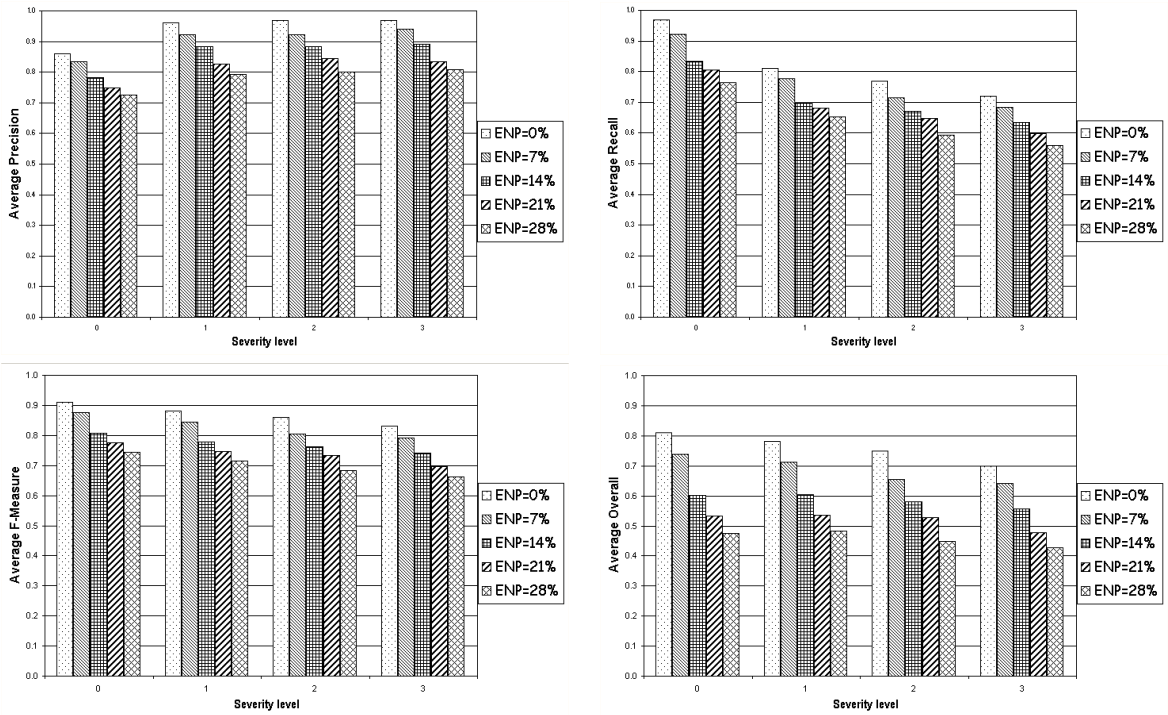

Figure 5: Average Precision, Average Recall, Average F-Measure and Average Overall for various values of ENP

Case	*Average Precision* *Severity Levels 0-1-2-3*	*Average Recall* *Severity Levels 0-1-2-3*	*Average F-Measure* *Severity Levels 0-1-2-3*	*Average Overall* *Severity Levels 0-1-2-3*
No errors	0.86 - 0.96 - 0.97 - 0.97	0.97 - 0.81 - 0.77 - 0.72	0.91 - 0.88 - 0.86 - 0.83	0.81 - 0.78 - 0.75 - 0.70
(a)	0.86 - 0.96 - 0.97 - 0.97	0.92 - 0.77 - 0.73 - 0.68	0.89 - 0.86 - 0.83 - 0.80	0.77 - 0.74 - 0.71 - 0.66
(b)	0.86 - 0.96 - 0.97 - 0.97	0.86 - 0.72 - 0.68 - 0.64	0.86 - 0.82 - 0.80 - 0.77	0.72 - 0.69 - 0.66 - 0.62
(c)	0.87 - 0.97 - 0.98 - 0.98	0.77 - 0.64 - 0.61 - 0.57	0.82 - 0.77 - 0.75 - 0.72	0.65 - 0.62 - 0.60 - 0.56
(d)	0.87 - 0.97 - 0.98 - 0.98	0.68 - 0.57 - 0.54 - 0.50	0.76 - 0.71 - 0.69 - 0.66	0.57 - 0.55 - 0.53 - 0.49
(e)	0.82 - 0.91 - 0.92 - 0.92	0.97 - 0.81 - 0.77 - 0.72	0.89 - 0.86 - 0.84 - 0.81	0.75 - 0.73 - 0.71 - 0.66
(f)	0.76 - 0.85 - 0.86 - 0.86	0.97 - 0.81 - 0.77 - 0.72	0.85 - 0.83 - 0.81 - 0.78	0.67 - 0.67 - 0.64 - 0.60
(g)	0.68 - 0.76 - 0.77 - 0.77	0.98 - 0.81 - 0.77 - 0.72	0.80 - 0.79 - 0.77 - 0.75	0.52 - 0.56 - 0.54 - 0.51
(h)	0.60 - 0.67 - 0.68 - 0.68	0.98 - 0.82 - 0.78 - 0.73	0.75 - 0.74 - 0.72 - 0.70	0.33 - 0.42 - 0.41 - 0.38

Table 5: Variation of Precision, Recall, F-Measure and Overall w.r.t. possible errors in the input thesauruses

suitable similarity functions; *(ii)* both of them are *schema-based*; *(iii)* both of them are *hybrid*; *(iv)* both of them exploit a thesaurus as *auxiliary information*. As for differences, we may observe that: *(i)* Cupid is based on tree matching whereas our approach is based on graph matching; *(ii)* Cupid is capable of managing *generic* data sources whereas our approach has been developed for operating only on XML sources; *(iii)* Cupid is capable of extracting also *1:n matchings* whereas our approach has been conceived for deriving only *1:1 matchings*.

MOMIS. Some similarities exist between our approach and MOMIS; in fact: *(i)* both of them are *schema-based*; *(ii)* both of them are *hybrid*; *(iii)* both of them derive *1:1 matchings*; *(iv)* both of them exploit a thesaurus as *auxiliary information*. As for differences, we may observe that: *(i)* MOMIS is based on description logics whereas our approach is graph-based; *(ii)* MOMIS is *generic*; *(iii)* MOMIS has been conceived mainly for integration and querying whereas our approach is specialized for inter-schema property extraction.

Approach of [14]. There exist some similarities between the approach of [14] and ours; specifically, *(i)* both of them are *schema-based*; *(ii)* both of them are *hybrid*; *(iii)* both of them derive *1:1 matchings*. There are also important differences between the two approaches; specifically: *(i)* in order to perform matching activities, the approach of [14] adopts *statistical-based* techniques whereas our approach operates on graphs; *(ii)* the approach of [14] operates on databases that can be accessed through Web query interfaces; *(iii)* the approach of [14] does not exploit *auxiliary information*; *(iii)* the approach of [14] creates a hidden schema which is both capable of fully describing a domain and useful as a mediated schema; such a characteristic is not present in our approach; however, as claimed by the authors, this makes the approach of [14] to be exponential; as a consequence, the approach of [14] can be applied only if schema match-

ing is carried out off-line; on the contrary, our approach is much more light and can be applied both on-line and off-line.

Approach of [4]. The main goal of the approach proposed in [4] is clearly different from that of our approach; in fact, the approach of [4] has been conceived to determine if data stored in an XML document approximatively conform to a DTD; by contrast, our approach aims to detect semantic similarities between two XML Schemas. Despite this substantial difference, we can observe that the approach of [4] and ours share some similarities. Specifically: *(i)* in both of them the analysis of structural properties of input data sources plays a key role; *(ii)* both of them clearly distinguish the roles played by simple and complex elements; *(iii)* both of them consider the constraints related to the occurrences of an element (e.g., if an element is optional or mandatory); *(iv)* both of them are specific for XML sources; *(v)* both of them are *hybrid*. As for the main differences between the two approaches, we observe that: *(i)* the approach of [4] is based on tree matching whereas our approach is based on graph matching; *(ii)* the approach of [4] is both *schema-based* and *instance-based*; *(iii)* the approach of [4] can extract *1:1*, *1:m* and *m:n matchings*; *(iv)* the approach of [4] does not exploit any *auxiliary information*.

DIKE. There are some similarities between our approach and DIKE; specifically, both of them: *(i)* are *graph-based*; *(ii)* are *schema-based*; *(iii)* are *hybrid*; *(iv)* exploit a thesaurus as *auxiliary information*.

However, there are also important differences between them; specifically: *(i)* DIKE operates on E/R schemas whereas our approach is graph-based. *(ii)* DIKE derives *1:1*, *1:n* and *m:n* matchings. *(iii)* The algorithms underlying DIKE rely on various thresholds and weights. *(iv)* DIKE does not consider a "severity" level that, on the contrary, plays a key role in our approach. *(v)* As far as the property derivation technique is concerned, DIKE and our system follow very different philosophies. As a matter of fact, DIKE exploits a sophisticated fixpoint computation strategy to derive interschema properties, whereas the approach we are presenting in this paper is simpler. *(vi)* Finally, the user intervention required by DIKE is heavier than that required by our approach since the former requires a tuning activity to be carried out for all thresholds and weights before the extraction process can start.

7 Conclusions

In this paper we have proposed an approach for the extraction of synonymies, hyponymies, overlappings and homonymies from a set of XML Schemas. We have shown that our approach is specialized for XML sources, is almost automatic, semantic and "light"; it derives all these properties in a uniform way and allows the choice of the "severity" level against which the extraction task must be performed.

We have illustrated some experiments that we have carried out to test its performance and to compare its results with those achieved by other approaches. We have also examined various related approaches previously proposed in the literature and we have compared them with ours from various points of views.

In the future we plan to investigate various research issues related to those presented here. First, we plan to develop approaches for deriving other typologies of interschema properties. Specifically, we would like to derive complex knowledge patterns involving a large variety of concepts belonging to different XML Schemas; in this application context we plan to exploit data mining techniques. After this, we would like to define new approaches for exploiting the properties considered in this paper, as well as those we shall study in the future, in the various application contexts where interschema properties can generally play a key role.

Finally, we would like to put the system described here as a part of a more complex system whose purpose is the extraction of intensional knowledge from semantically heterogeneous XML sources and its exploitation for handling their interoperability.

References

[1] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.

[2] J. Berlin and A. Motro. Autoplex: Automated discovery of content for virtual databases. In *Proc. of the International Conference on Cooperative Information Systems (CoopIS 2001)*, pages 108–122, Trento, Italy, 2001. Lecture Notes in Computer Science, Springer.

[3] J. Berlin and A. Motro. Database schema matching using machine learning with feature selection. In *Proc. of the International Conference on Advanced Information Systems Engineering (CAiSE 2002)*, pages 452–466, Toronto, Canada, 2002. Lecture Notes in Computer Science, Springer.

[4] E. Bertino, G. Guerrini, and M. Mesiti. A matching algorithm for measuring the structural similarity between an XML document and a DTD and its applications. *Information Systems*, 29(1):23–46, 2004.

[5] P. De Meo, G. Quattrone, G. Terracina, and D. Ursino. Integration of XML Schemas at various "severity" levels. *Information Systems*, 31(6):397–434, 2006.

[6] H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proc. of the International Workshop on Web, Web-Services, and Database Systems*, pages 221–237, Erfurt, Germany, 2002. Lecture Notes in Computer Science, Springer.

[7] H. Do and E. Rahm. COMA- a system for flexible combination of schema matching approaches. In *Proc. of the International Conference on Very Large Databases (VLDB 2002)*, pages 610–621, Hong Kong, China, 2002. VLDB Endowment.

[8] A. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: a machine-learning approach. In *Proc. of the ACM International Conference on Management of Data (SIGMOD 2001)*, pages 509–520, Santa Barbara, California, USA, 2001. ACM Press.

[9] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to match ontologies on the Semantic Web. *The International Journal on Very Large Databases*, 12(4):303–319, 2003.

[10] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the Semantic Web. In *Proc. of the ACM International Conference on World Wide Web (WWW 2002)*, pages 662–673, Honolulu, Hawaii, USA, 2002. ACM Press.

[11] A. Gal, A. Anaby-Tavor, A. Trombetta, and D. Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *The International Journal on Very Large Databases*, 14(1):50–67, 2005.

[12] Z. Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 18:23–38, 1986.

[13] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an Algorithm and an Implementation of Semantic Matching. In *Proc. of the European Semantic Web Symposyium (ESWS'04)*, pages 61–75, Heraklion, Crete, Greece, 2004. Springer, Lecture Notes in Computer Science.

[14] B. He and K. Chen-Chuan Chang. Statistical schema matching across Web query interfaces. In *Proc. of the ACM International Conference on Management of Data (SIGMOD 2003)*, pages 217–228, San Diego, California, United States, 2003. ACM Press.

[15] M.L. Lee, L.H. Yang, W. Hsu, and X. Yang. XClust: clustering XML schemas for effective integration. In *Proc. of the ACM International Conference on Information and Knowledge Management (CIKM 2002)*, pages 292–299, McLean, Virginia, USA, 2002. ACM Press.

[16] W. Li and C. Clifton. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data and Knowledge Engineering*, 33(1):49–84, 2000.

[17] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In *Proc. of the International Conference on Very Large Data Bases (VLDB 2001)*, pages 49–58, Roma, Italy, 2001. Morgan Kaufmann.

[18] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity Flooding: A versatile graph matching algorithm and its application to schema matching. In *Proc. of the IEEE International Conference on Data Engineering (ICDE 2002)*, pages 117–128, San Jose, California, USA, 2002. IEEE Computer Society Press.

[19] A.G. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[20] L. Palopoli, D. Saccà, G. Terracina, and D. Ursino. Uniform techniques for deriving similarities of objects and subschemes in heterogeneous databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):271–294, 2003.

[21] L. Palopoli, G. Terracina, and D. Ursino. Experiences using DIKE, a system for supporting cooperative information system and data warehouse design. *Information Systems*, 28(7):835–865, 2003.

[22] K. Passi, L. Lane, S.K. Madria, B.C. Sakamuri, M.K. Mohania, and S.S. Bhowmick. A model for XML Schema integration. In *Proc. of the International Conference on E-Commerce and Web Technologies (EC-Web 2002)*, pages 193–202, Aix-en-Provence, France, 2002. Lecture Notes in Computer Science, Springer.

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

[24] V. C. Storey. Understanding semantic relationships. *The International Journal on Very Large Databases*, 2(4):455–488, 1993.

[25] C.J. Van Rijsbergen. *Information Retrieval*. Butterworth, 1979.