Deriving synonymies and homonymies of object classes in semi-structured information sources

Giorgio Terracina
D.E.I.S., Università della Calabria
Via Pietro Bucci
87036 Rende (CS), Italy
terracina@si.deis.unical.it

Domenico Ursino
D.E.I.S., Università della Calabria
Via Pietro Bucci
87036 Rende (CS), Italy
ursino@si.deis.unical.it

ABSTRACT

The problem of handling both the integration and the cooperation of a large number of semi-structured information sources is today a challenging issue. In this context, a central role can be played by the knowledge about the semantic relationships between object classes belonging to different semi-structured information sources (inter-source properties). In this paper we propose a semi-automatic approach for extracting two kinds of inter-source properties, namely synonymies and homonymies, from different semi-structured information sources. In order to carry out the extraction task, we introduce both a conceptual model for representing object classes belonging to semi-structured information sources and a metrics for measuring the strength of the semantic relationships between object classes belonging to the same semi-structured information source.

1. INTRODUCTION

1.1 Motivations

Nowadays, much of the existing electronic data, such as HTML, SGML and XML documents, legacy systems and textual documents, are "non-relational" and, as such, they cannot be handled by database systems. Indeed, the common characteristic of all data handled by DBMS consists in having a well defined structure. On the contrary, the other forms of electronic data generally have not a precise, previously known structure and, even when this is known, it may change frequently [1, 2, 3, 7, 8, 10, 12, 15, 20, 21, 22, 23]. Structured information sources, such as databases, can be considered as a particular (and simpler) case of semi-structured information sources. As a consequence, it appears reasonable that some of the ideas applied to handle database schemes can be used also for semi-structured data. Obviously they can be exploited only as leading ideas since the context is much more general.

An interesting idea exploited towards obtaining heterogeneous database interoperability consists in extracting *inter-source properties*, i.e., terminological and structural properties holding between objects of different schemes; these can be exploited above all for defining the semantics of involved schemes and then for integrating heterogeneous information sources, for designing techniques for converting data, for optimization and querying, and so on [4, 6, 9, 16, 17, 18, 24]. These are also some of the challenging research issues for semi-structured data [22].

In this paper we propose a semi-automatic technique for deriving some inter-source properties between object classes belonging to different semi-structured information sources. In particular we focus on the extraction of synonymies and homonymies; a synonymy between two object classes C_1 and C_2 indicates that they represent the same concept; an homonymy between two object classes C_1 and C_2, having the same name, indicates that they represent different concepts. In order to carry out this task, we need to define: *(i)* a new conceptual model well suited for representing object classes belonging to semi-structured information sources, *(ii)* a suitable metrics, based on this model, for measuring the strength of the semantic relationships between object classes belonging to the same semi-structured information source.

In this paper we assume that the semi-structured information sources we consider are represented using the Object Exchange Model (OEM) [1, 2, 7, 20]. In fact, this is one of the most common models for representing semi-structured data. Note that this is not an actual limitation, since rules can be defined for obtaining the representation of other kinds of semi-structured information sources (such as XML documents) in terms of the conceptual model we present in this paper [19].

1.2 General characteristics of the approach

Our conceptual model is obtained by associating to each information source S a network $Net(S)$, called *SDR-Network* (Semantic Distance-Relevance Network). $Net(S)$ has a node for each class in S. Arcs in $Net(S)$ are labeled: given an arc from x to y, the label l_{xy} consists of a pair $[d_{xy}, r_{xy}]$, where d_{xy} denotes the *semantic distance* between x and y, i.e. the capability of the concept associated to y to characterize the concept associated to x, whereas r_{xy} denotes the *semantic relevance* of y w.r.t. x, i.e. the participation degree of the

ADVANCES IN DATA MANAGEMENT 2000
Krithi Ramamritham, T.M. Vijayaraman (Editors)
Tata McGraw-Hill Publishing Company Ltd.

concept associated to y in defining the concept associated to x (see Section *3.1.4* for a formal definition of both the semantic distance and the semantic relevance).

The *technique for extracting synonymies and homonymies* takes in input a set of semi-structured information sources and returns a Synonymy Dictionary SD and an Homonymy Dictionary HD. Derived properties are fuzzy and are represented as triplets $\langle A, B, f \rangle$, where A and B are the involved object classes and f is a fuzzy coefficient, in the real interval $[0, 1]$, expressing the plausibility of the property.

The proposed tehcnique takes some ideas from [17] where we proposed the analysis of object neighborhoods for deriving synonymies and homonymies in database schemes. In more detail, in order to derive the similarity between two object classes C_1 and C_2, we analyze both the two object classes and their neighborhoods. The influence of the similarity between the neighborhoods of C_1 and C_2 on the similarity of C_1 and C_2 depends on the closeness of neighborhoods; in particular the closer to C_1 and C_2 the neighborhoods are, the strongest the influence is. In order to formalize this concept, we define the *i_th* neighborhood of a class x, denoted by $nbh(x, i)$, as the set of arcs in $Net(S)$ whose target node has a semantic distance from x greater than or equal to i and lesser than $i + 1$. A monotone decreasing weighting succession $\{p(i)\}$ is associated to neighborhoods of x so that farthest neighborhoods will have lightest weights.

Intuitively, our approach for the detection of synonymies and homonymies relative to two semi-structured information sources S_1 and S_2 consists of the following steps:

- Constructing a set of basic similarities; these are rough properties taking into account only lexical similarities and the nearest neighborhoods; they are to be considered as the starting point for the extraction of "real" properties.
- Visiting, for each pair of classes $C_l \in S_1$ and $C_m \in S_2$, $nbh(C_l, i)$ and $nbh(C_m, i)$.
- Computing the similarity degree between $nbh(C_l, i)$ and $nbh(C_m, i)$ as an objective function associated to the maximum weight matching algorithm on a suitable bipartite weighted graph defined from classes of both $nbh(C_l, i)$ and $nbh(C_m, i)$.
- Computing the overall similarity degree of C_l and C_m as a weighted mean of similarity degrees between the various neighborhoods of C_l and C_m; weights are the elements of the succession $\{p(i)\}$ described above.
- Obtaining synonymies (resp., homonymies) from derived similarities by taking those ones having a plausibility coefficient greater than (resp., lesser than) a certain, dynamically computed threshold th_{Syn} (resp., th_{Hom}).

1.3 Plan of the paper

The plan of the paper is as follows. In Section 2 we describe the OEM model. Section 3 is devoted to present the conceptual model and the related metrics. The derivation of synonymies and homonymies, as well as the application of the proposed technique to a real example case, is shown in Section 4. Finally, in Section 5, we draw our conclusions.

2. THE OBJECT EXCHANGE MODEL

The Object Exchange Model (hereafter OEM) [1, 2, 7, 15, 20] is one of the most common models used for representing semi-structured data. Data described in a semi-structured information source are associated to objects in the OEM. Each object has a unique *object identifier (oid)* whose value belongs to the type `oid`.

There are two kinds of possible objects, namely Atomic Objects and Complex Objects. Concepts associated to *Atomic Objects* are described by a single value, taken from one of the disjoint basic atomic types, e.g., `integer`, `real`, `string`, `gif`, `html`, `audio`, `java`, etc. All non-atomic objects are *Complex Objects*; they are specified by a set of object references. Each object reference has the form *(label, oid)*, in which *label* describes the reference and *oid* specifies the referred object. The domain of the labels is the atomic type `string`.

A representation in the OEM basically consists of a labeled rooted graph. Each node of this graph represents an (atomic or complex) object; each arc is associated to a reference; in particular, an arc from a node N_S to a node N_T having a label l represents the reference (l, N_T) specifying N_S.

An *OEM-Graph* can be represented as:

$$\langle N_A^{OEM}(S) \cup N_C^{OEM}(S), A^{OEM}(S) \rangle$$

where $N_A^{OEM}(S)$ is the set of atomic nodes, $N_C^{OEM}(S)$ denotes the set of complex nodes and $A^{OEM}(S)$ indicates the set of arcs.

An example of an OEM-Graph is illustrated in Figure 1. Note that leaf nodes correspond to atomic objects whereas complex objects are represented by the other nodes.

3. THE CONCEPTUAL MODEL AND THE RELATED METRICS

The construction of a *conceptual model* and of a *related metrics* for measuring semantic distances and relevances of object classes belonging to an information source is much more difficult in the semi-structured than in the structured case. As a matter of fact, the difficulties are both syntactic and semantic.

In particular, *syntactic difficulties* are due to the fact that structured information sources can be represented by the E/R model; this is simple, complete and commonly accepted. Vice versa models for representing semi-structured data are more complex and various [1, 2, 5, 7, 13, 14, 15, 20]; as an example, the OEM is based on concepts as graphs, nodes, labels, arcs; these concepts are different from those characterizing the E/R model. In addition the OEM represents instances, i.e. extensional data, whereas the E/R model represents object classes, i.e. intensional data. Since the metrics is referred to object classes, the corresponding conceptual model we propose must consider intensional data and the necessity arises to derive object classes associated to instances represented in the OEM.

Semantic difficulties arise since, in semi-structured information sources, the various objects of the same class can be described by different properties. In other words, a property can be present in some objects of a class whereas can be absent from other ones. As a consequence we have the

need to take into account how often a property participates in the definition of the objects of a class.

3.1 The SDR-Network

In this section we formally introduce the *SDR-Network* $Net(S)$ associated to a semi-structured information source S.

Given a source S the associated *SDR-Network* $Net(S)$ is:

$$Net(S) = \langle N^{SDR}(S), A^{SDR}(S) \rangle$$

where $N^{SDR}(S)$ represents a set of nodes and $A^{SDR}(S)$ denotes a set of arcs. In more detail, each node is characterized by a name; each arc can be represented by a triplet $\langle x, y, l_{xy} \rangle$, where x is the source, y is the target and l_{xy} is a label associated to the arc. l_{xy} can be represented, in its turn, as a pair $[d_{xy}, r_{xy}]$, where both d_{xy} and r_{xy} belong to the real interval $[0, 1]$. We call d_{xy} the *semantic distance coefficient*; it indicates the capability of the concept associated to y to characterize the concept associated to x. We call r_{xy} the *semantic relevance coefficient*; it denotes the participation degree of the concept expressed by y in the definition of the concept associated to x; the precise semantics of these coefficients and their derivation are described in Section *3.1.4*.

Before illustrating how a SDR-Network can be constructed from a corresponding OEM-Graph, it is worth pointing out that, in an OEM-Graph, nodes represent objects whereas, in a SDR-Network, nodes are associated to classes of objects. Therefore, from now on, we use the term *object* to indicate the object associated to an OEM node whereas we use the term *class* to indicate the class of objects represented by a SDR-Network node.

The process of constructing a SDR-Network from the corresponding OEM-Graph consists of four phases: *(i)* pre-processing: it basically creates a modified OEM-Graph where a unique name is associated to each node; *(ii)* definition of the SDR-Network nodes; *(iii)* definition of the SDR-Network arcs; *(iv)* definition of labels associated to SDR-Network arcs. In the following subsections we describe these phases in more detail; two examples of the construction of the SDR-Network associated to an OEM-Graph are also given.

3.1.1 The pre-processing phase

The pre-processing phase is intended to obtain a modified OEM-Graph in which each node has an associated name.

Generally, the name associated to each node is inherited from the label of the arc which the considered node is target of. However two particular cases must be considered:

1. If an atomic node t exists being the target of more than one arc, the pre-processing phase creates a new target node t_i for each arc a_i incident onto t; the name of t_i is inherited from the label of a_i.

2. If a complex node t exists being the target of more than one arc and some of these have different labels, the necessity arises to determine what are the significant roles associated to t; this task is carried out with the support of human experts, which must decide if either the involved labels have the same meaning or one has a more specific meaning than the other or they have different meanings. In the first case only one node is maintained and the user must choose the label which best characterizes it; in the second and in the third case the user must choose if either the node must be duplicated (in which case names associated to obtained nodes are inherited from the corresponding labels) or if only one node must be maintained (in which case the user must choose, as the name of the node, the label which best characterizes it).

In the following we assume that, when referring to an OEM-Graph, we intend the OEM-Graph modified by the pre-processing phase.

3.1.2 Definition of the SDR-Network nodes

The set $N^{SDR}(S)$ of nodes in the SDR-Network is actually composed by the union of two sets of nodes:

$$N^{SDR}(S) = N_C^{SDR}(S) \cup N_A^{SDR}(S)$$

$N_C^{SDR}(S)$ is the set of nodes in the SDR-Network derived from complex nodes in the OEM-Graph. In particular, for each set of complex nodes in the OEM-Graph having the same name M, a node of $N_C^{SDR}(S)$ is associated to them whose name is M. Nodes belonging to $N_C^{SDR}(S)$ constitute the set of *complex nodes* of $Net(S)$.

$N_A^{SDR}(S)$ is the set of nodes in the SDR-Network derived from atomic nodes in the OEM-Graph. In particular, for each set of atomic nodes in the OEM-Graph such that they have the same name M and there does not exist a complex node in the OEM-Graph whose name is M, a node of $N_A^{SDR}(S)$, named M, is associated to them. Nodes belonging to $N_A^{SDR}(S)$ form the set of *atomic nodes* of $Net(S)$.

3.1.3 Definition of the SDR-Network arcs

Before introducing rules for obtaining SDR-Network arcs from the corresponding OEM-Graph, the following definitions are needed.

Definition 1. An *OEM-arc* is an arc in an OEM-Graph. It can be represented by a triplet $\langle S, T, L \rangle$, where S is the source node, T is the target node and L is the corresponding label. A *SDR-arc* is an arc in a SDR-Network. It can be represented by a triplet $\langle S, T, L \rangle$, where S is the source node, T is the target node and L is the corresponding label. □

Definition 2. Let G_{OEM} be an OEM-Graph and Net_{SDR} be the corresponding SDR-Network. Let N_G be a node of G_{OEM}. The *SDR-Corr-Node* of N_G is the node N_N of Net_{SDR} corresponding to N_G. The *OEM-Corr-NodeSet* of N_N is the set of nodes of G_{OEM} which N_N is derived from. □

A function $SDR\text{-}Corr\text{-}Node(N_G)$ is defined, which takes in input a node N_G of an OEM-Graph and yields in output the SDR-Corr-Node N_N of N_G. A function $OEM\text{-}Corr\text{-}NodeSet(N_N)$ is defined, which takes in input a node N_N of a SDR-Network and returns the OEM-Corr-NodeSet NS_G of N_N.

We are now able to define the rules for obtaining arcs of a SDR-Network from the corresponding OEM-Graph. In particular, all the OEM-arcs $\langle S_i, T_i, L \rangle$ such that, for each i, the SDR-Corr-Node of S_i is a unique node N_S and the

SDR-Corr-Node of T_i is a unique node N_T, are represented by an arc from N_S to N_T. Note that an OEM-arc $\langle S, T, L\rangle$ such that the SDR-Corr-Node of S is equal to the SDR-Corr-Node of T produces a cyclic arc in the SDR-Network.

3.1.4 Definition of labels associated to arcs of the SDR-Network

Let Net_{SDR} be a SDR-Network and let $\langle S, T, L\rangle$ be one of its SDR-arcs. We have seen that the label L can be represented by a pair of values $[d_{ST}, r_{ST}]$, where d_{ST} is the semantic distance coefficient and r_{ST} is the semantic relevance coefficient.

In order to compute d_{ST} and r_{ST} from the nodes, the arcs and the labels associated to the corresponding OEM-Graph, we must first define the following sets of nodes: *(i)* NS_S denotes the OEM-Corr-NodeSet of S; *(ii)* NS_T indicates the OEM-Corr-NodeSet of T; *(iii)* $RNS_{S,T}$ represents the set of nodes n_i such that, for each i, $n_i \in NS_S$ and there exists at least one node $q \in NS_T$ such that an OEM-arc $\langle n_i, q, l_{n_i q}\rangle$ is present in the corresponding OEM-Graph.
We are now able to formally define d_{ST} and r_{ST}. In particular, as for the *semantic distance coefficient*, we have:

$$d_{ST} = \frac{\sum_{n_i \in RNS_{S,T}} \gamma(n_i, T)}{|RNS_{S,T}|}, \qquad \text{where}$$

$$\gamma(n_i, T) = \begin{cases} 0 & \text{if } \exists\langle n_i, p, l_{n_i p}\rangle \text{ such that } p \in NS_T, \\ & p \text{ is an atomic node and} \\ & \nexists\langle n_i, q, l_{n_i q}\rangle \text{ such that } q \in NS_T, \\ & q \neq p \\ 0.5 & \text{if } \exists\langle n_i, p, l_{n_i p}\rangle \text{ and } \exists\langle n_i, q, l_{n_i q}\rangle \\ & \text{such that } p, q \in NS_T, p \neq q \text{ and} \\ & p, q \text{ are atomic nodes and} \\ & \nexists\langle n_i, r, l_{n_i r}\rangle \text{ such that } r \in NS_T, \\ & \text{r is a complex node} \\ 1 & \text{if } \exists\langle n_i, p, l_{n_i p}\rangle \text{ such that } p \in NS_T \\ & \text{and } p \text{ is a complex node} \end{cases}$$

The reasoning underlying the definition of d_{ST} is as follows: an atomic OEM-node defines directly the concept associated to the corresponding atomic object; a complex OEM-node defines the concept associated to the corresponding object by means of the set of its references. Thus, given an OEM-node N_O (belonging to $RNS_{S,T}$), an atomic OEM-node N'_O (belonging to NS_T and connected to N_O) is semantically closer to N_O than a complex OEM-node N''_O (belonging to NS_T and connected to N_O) because it does not need the support of further nodes for defining a property of N_O. Moreover, if two or more atomic OEM-nodes with the same name are linked to the same OEM-node N_O, we can conclude that one of them alone is not enough to completely specify a given property of N_O whereas they, as a whole, do specify this property. In this case the semantic distance between each of these atomic nodes and N_O is intermediate w.r.t. the distances defined above.
As far as the *semantic relevance coefficient* is concerned, we have:

$$r_{ST} = \frac{|RNS_{S,T}|}{|NS_S|}$$

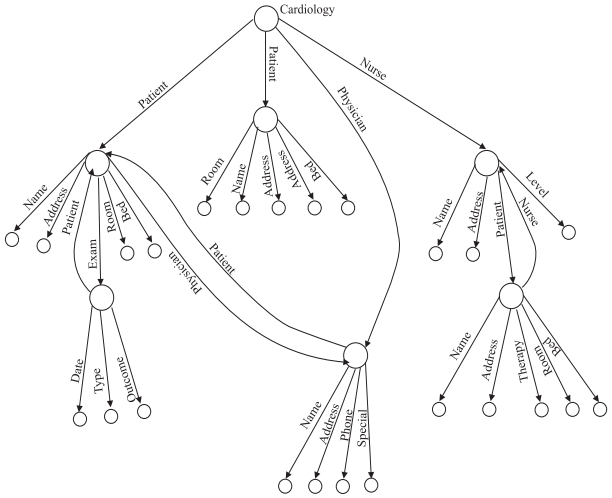

Figure 1: The OEM-Graph of a Cardiology Division of an hospital

This formula directly derives from the definition of the semantic relevance of T w.r.t. S as the participation degree of the concept associated to T in defining the concept associated to S.

3.1.5 Examples

Consider the OEM-Graph shown in Figure 1, relative to a Cardiology Division of an hospital, and the corresponding SDR-Network shown in Figure 2 (example taken from [6]).
The following considerations can be drawn:

- All nodes of the OEM-Graph having an incident arc with label *Patient* are represented, in the corresponding SDR-Network, by a unique node having name *Patient*. The same reasoning has been applied for obtaining both nodes *Physician*, *Exam* and *Nurse* of the SDR-Network as well as those nodes of the SDR-Network derived from atomic nodes of the OEM-Graph.

- The SDR-arc between nodes *Patient* and *Physician* corresponds to all OEM-arcs $\langle S_i, T_i, L\rangle$ such that, for each i, the SDR-Corr-Node of S_i is the node *Patient* and the SDR-Corr-Node of T_i is the node *Physician*. In the same way all the other SDR-arcs are obtained.

- $d_{Patient,Physician}$ is 1 since *(i)* $NS_{Patient}$ is composed by three nodes, *(ii)* $NS_{Physician}$ is composed by one node, *(iii)* $RNS_{Patient,Physician}$ is composed by the unique node n_p of $NS_{Patient}$ linked by an arc to the unique node belonging to $NS_{Physician}$, *(iv)* the value of $\gamma(n_p, Physician)$ is 1 since the node belonging to $NS_{Physician}$ is complex. All the other semantic distance coefficients are obtained in the same way.

- For the computation of $r_{Patient,Physician}$ we observe that $|RNS_{Patient,Physician}| = 1$, $|NS_{Patient}| = 3$, therefore $r_{Patient,Physician} = 0.33$. $r_{Physician,Patient} = 1$ because $|RNS_{Physician,Patient}| = 1$ and $|NS_{Physician}| = 1$. It is worth pointing out that $r_{Patient,Physician} \neq r_{Physician,Patient}$. An analogous reasoning allows to obtain all the other semantic relevance coefficients.

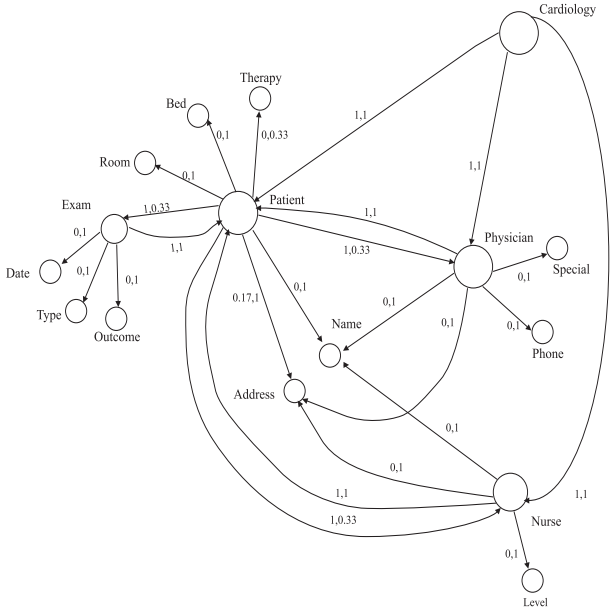

Figure 2: The SDR-Network corresponding to the OEM-Graph of Figure 1

As a further example of the construction of the SDR-Network from an OEM-Graph consider the OEM-Graph shown in Figure 3, relative to a Restaurant Guide (this example has been derived from [1, 2]), and the corresponding SDR-Network depicted in Figure 4.

The following considerations can be drawn:

- In the OEM-Graph there are two arcs having the same label *Price* being incident on the same target atomic node; in this case, during the pre-processing phase, the target node is duplicated and both nodes have name *Price*.
- In the OEM-Graph there are some nodes which are the target nodes of arcs having both *Restaurant* and *Nearby* as labels. In this case, during the pre-processing phase, the intervention of a human expert is required; this determines that the two labels have different meanings; however he judges that there is no necessity to duplicate the corresponding nodes since *Nearby* represents a relationship between two objects and not a concept of its own. Therefore the name *Restaurant* is associated to these nodes.
- After the pre-processing phase, in the OEM-Graph, there is a set of atomic nodes such that *(i)* they have the same name *Address* and *(ii)* in the OEM-Graph a complex node N_C *exists* whose name is *Address*. In this situation, in the SDR-Network, all these atomic nodes are represented by *SDR-Corr-Node*(N_C).

3.2 Definition of the metrics

We are now in the position of establishing our metrics for measuring the semantic distance and the semantic relevance between two classes C and C' within an Information Source

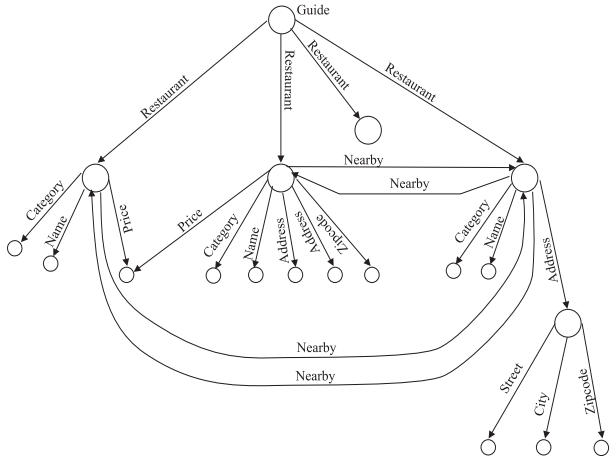

Figure 3: The OEM-Graph of a Restaurant Guide

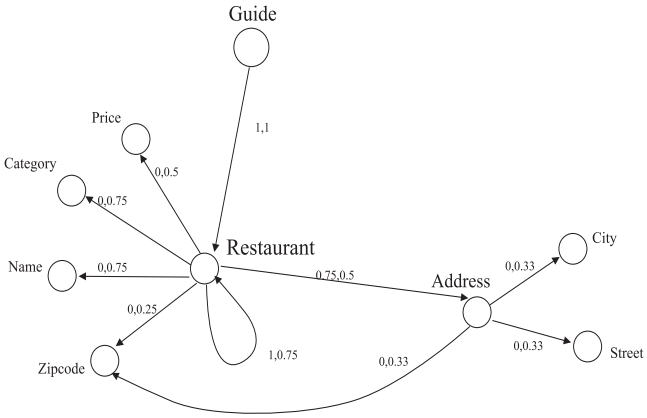

Figure 4: The SDR-Network corresponding to the OEM-Graph of Figure 3

S. Before describing our metrics we must introduce the following support functions:

- $ClassOf(N) \rightarrow C$, which takes in input a node N of the *SDR-Network* $Net(S)$ associated to an information source S and yields in output the class C of S corresponding to N;
- $NodeOf(C) \rightarrow N$, which takes in input a class C of an information source S and yields in output the node N of the associated *SDR-Network* $Net(S)$ corresponding to C.

Now, consider the following definitions:

Definition 3. Define the *Path Semantic Distance* of a path P in $Net(S)$ (denoted by PSD_P) as the sum of the semantic distance coefficients associated to the arcs constituting the path. Define the *Path Semantic Relevance* of a path P in $Net(S)$ (denoted by PSR_P) as the product of the semantic relevance coefficients associated to the arcs constituting the path. □

Definition 4. The *D_Shortest_Path* between two nodes N and N' in $Net(S)$ (denoted by $\lfloor N, N' \rfloor$) is the path having the minimum Path Semantic Distance among the paths connecting N and N'. If more than one path exists having the same minimum Path Semantic Distance, one of those having the maximum Path Semantic Relevance is chosen. Define the *CD_Shortest_Path* (Conditional D_Shortest_Path) between two nodes N and N' in $Net(S)$ and including an arc A (denoted by $\lfloor N, N' \rfloor_A$) as the path having the minimum Path Semantic Distance among those connecting N and N' and including A. If more than one path exists having the same minimum Path Semantic Distance, one of those having the maximum Path Semantic Relevance is chosen. □

Definition 5. Define a D_Path_n as a path P in $Net(S)$ such that $n \leq PSD_P < n+1$. □

Definition 6. Define the *i_th neighborhood* of a class x as:

$$nbh(x,i) = \{A | A \in A^{SDR}(S), A = \langle z, y, l_{zy} \rangle, \lfloor NodeOf(x), y \rfloor_A \text{ is a } D\_Path_i, NodeOf(x) \neq y, (\forall j < i)(A \notin nbh(x,j))\} \quad i \geq 0$$

□

Thus, an arc $\langle z, y, l_{zy} \rangle$ belongs to $nbh(x,i)$ if it does not belong to any neighborhood lesser than i and there exists a CD_Shortest_Path from $NodeOf(x)$ to y including $\langle z, y, l_{zy} \rangle$ which is a D_Path_i. Note that the possibility exists that $NodeOf(x) = z$.

PROPOSITION 3.1. *Let x be a class. Then, for each $i > 0$, $nbh(x,i) = \emptyset$ implies that $((\forall j > i)(nbh(x,j) = \emptyset))$.*

PROOF. Immediate from the definitions of neighborhood and of D_Path$_i$ and by noting that the semantic distance coefficient associated to an arc is lesser than or equal to 1. □

PROPOSITION 3.2. *Let x be a class and let $\bar{i} > 0$ be the maximum integer such that $nbh(x,\bar{i}) \neq \emptyset$; then $\bigcup_{0 \leq j \leq (k-1)} nbh(x,j) \subset \bigcup_{0 \leq j \leq k} nbh(x,j)$ for each k such that $0 < k \leq \bar{i}$.*

PROOF. Immediate from Proposition 3.1 and by noting that an arc belongs to $nbh(x,i)$ only if it does not belong to $nbh(x,j)$ for all $j < i$. □

3.2.1 Example

Consider the node *Exam* of the *SDR-Network* shown in Figure 2[1]. We have that:

$$nbh(Exam,0) = \{\langle Exam, Date, [0,1] \rangle, \langle Exam, Type, [0,1] \rangle, \langle Exam, Outcome, [0,1] \rangle\}$$

Indeed the first arc belongs to $nbh(Exam,0)$ because $\lfloor Exam, Date \rfloor_{\langle Exam, Date, [0,1] \rangle}$ is a D_Path_0 and $Exam \neq Date$. An analogous reasoning can be done for the other two arcs.

[1]As stated above, there is a complete correspondence between a class of an information source S and a node of $Net(S)$; therefore, in order to simplify the notation, we use the same name (e.g., *Exam*) for indicating the class and the SDR-Network node corresponding to this class.

$$nbh(Exam,1) = \{\langle Exam, Patient, [1,1] \rangle, \langle Patient, Therapy, [0,0.33] \rangle, \langle Patient, Bed, [0,1] \rangle, \langle Patient, Room, [0,1] \rangle, \langle Patient, Address, [0.17,1] \rangle, \langle Patient, Name, [0,1] \rangle\}$$

Indeed the arc $A = \langle Patient, Therapy, [0,0.33] \rangle$ belongs to $nbh(Exam,1)$ because $\lfloor Exam, Therapy \rfloor_A$ is a D_Path_1, $Exam \neq Therapy$ and A does not belong to $nbh(Exam,0)$. The other arcs of $nbh(Exam,1)$ are obtained by an analogous reasoning. Finally the other non-empty neighborhoods of *Exam* are:

$$nbh(Exam,2) = \{\langle Patient, Physician, [1,0.33] \rangle, \langle Physician, Special, [0,1] \rangle, \langle Physician, Phone, [0,1] \rangle, \langle Physician, Name, [0,1] \rangle, \langle Physician, Address, [0,1] \rangle, \langle Patient, Nurse, [1,0.33] \rangle, \langle Nurse, Level, [0,1] \rangle, \langle Nurse, Name, [0,1] \rangle, \langle Nurse, Address, [0,1] \rangle\}$$

$$nbh(Exam,3) = \{\langle Physician, Patient, [1,1] \rangle, \langle Nurse, Patient, [1,1] \rangle\}$$

4. EXTRACTING SYNONYMIES AND HOMONYMIES

The process of extracting synonymies and homonymies between object classes belonging to semi-structured information sources takes in input both involved information sources (represented by the corresponding SDR-Networks) and some lexical synonymies between names stored in a Lexical Synonymy Property Dictionary $LSPD$. Lexical properties can be easily derived from a standard thesaurus (such as Wordnet); however, in order to obtain more precise results, we have associated a plausibility coefficient to each lexical property. In particular, domain experts are assumed to define the plausibility coefficient for each interesting lexical synonymy and, in order to obtain a high objectivity, the same coefficient is asked to more experts and the mean value is assumed.

In order to take into account the presence of errors occurred in the definition of plausibility properties we have conducted a sensitivity analysis based on varying the values specified for lexical similarities. The analysis has shown that, if errors are reasonable and do not involve most coefficients, the results yielded by our technique are not significantly influenced. Lexical properties are represented as triplets of the form $\langle A, B, f \rangle$, where A and B are class names and $f \in [0,1]$ indicates the plausibility of the property. Here and in the following we assume that two information sources S_1 and S_2 are given; the functions illustrated below have S_1, S_2 and $LSPD$ as implicit parameters.

The extraction of synonymies and homonymies can be carried out in two phases: the first phase derives the so called basic similarities which are rough properties taking into account only lexical similarities and the nearest neighborhoods; they are to be considered as the starting point for the extraction of "real" properties; this last operation is carried out by the second phase of our approach. The two phases are described in the next two subsections.

4.1 Phase 1: Derivation of basic similarities

This phase derives some basic similarities; these denote similitudes between information source classes; they are represented by triplets of the form $\langle C_1, C_2, f \rangle$, where C_1 and C_2 are the classes into consideration and f is a coefficient, in

the real interval $[0, 1]$, denoting the plausibility of the property; all basic similarities are stored in a *Basic Similarity Dictionary* (BSD).

Before describing how the BSD is obtained we must introduce the following support functions:

- $AN\_Set(N)$: it takes in input a node N of a *SDR-Network* $Net(S)$ and returns the set of atomic nodes connected to N by an outgoing arc;
- $C\_AN\_Set(N_1, N_2)$: it is a boolean function which takes in input two nodes N_1 and N_2 of a *SDR-Network* $Net(S)$ and returns *true* if both the conditions $(AN\_Set(N_1) \neq \emptyset)$ and $(AN\_Set(N_2) \neq \emptyset)$ are verified, *false* otherwise.

The derivation of BSD is carried out by the function η. For each pair of classes $C_1 \in S_1$ and $C_2 \in S_2$, the function η returns a tuple $\langle C_1, C_2, g_{C_1C_2}\rangle$, where $g_{C_1C_2}$ is a weighted mean of values returned by functions η_l, taking into account lexical similarities, and η_m, taking into consideration the similarity of the sets of atomic nodes directly connected to $NodeOf(C_1)$ in $Net(S_1)$ and $NodeOf(C_2)$ in $Net(S_2)$. More formally we have:

$$\begin{aligned} BSD = \eta() = \{\langle C_1, C_2, g_{C_1C_2}\rangle \mid\ & N_1 = NodeOf(C_1), \\ & N_2 = NodeOf(C_2), \\ & g_{C_1C_2} = \alpha_{lm}(N_1, N_2) \times \eta_l(N_1, N_2) + \\ & (1 - \alpha_{lm}(N_1, N_2)) \times \eta_m(N_1, N_2)\} \end{aligned}$$

4.1.1 Function α_{lm}.

The function $\alpha_{lm}(N_1, N_2)$ takes in input two nodes $N_1 \in Net(S_1)$ and $N_2 \in Net(S_2)$ and returns a value which is used to weigh the results obtained by functions η_l and η_m. The returned value depends on the presence of a lexical synonymy between the names of N_1 and N_2[2] and on the existence of atomic nodes directly connected to both N_1 and N_2. In particular α_{lm} can be defined as follows:

$$\alpha_{lm}(N_1, N_2) = \begin{cases} 0 & \text{if } \langle Name(N_1), Name(N_2), f_{N_1N_2}\rangle \notin LSPD \text{ and } C\_AN\_Set(N_1, N_2) \text{ returns true} \\ 0.5 & \text{if } \langle Name(N_1), Name(N_2), f_{N_1N_2}\rangle \in LSPD \text{ and } C\_AN\_Set(N_1, N_2) \text{ returns true} \\ 1 & \text{if } C\_AN\_Set(N_1, N_2) \text{ returns false} \end{cases}$$

Note that if either N_1 or N_2 are atomic nodes then $C\_AN\_Set(N_1, N_2)$ returns false and the basic similarity associated to N_1 and N_2 is a function of the lexical similarity alone.

4.1.2 Function η_l.

The function $\eta_l(N_1, N_2)$ takes in input two nodes $N_1 \in Net(S_1)$ and $N_2 \in Net(S_2)$ and returns a real value denoting the lexical synonymy between names of N_1 and N_2. The function can be encoded as follows:

$$\eta_l(N_1, N_2) = \begin{cases} g_{N_1N_2} & \text{if } \langle Name(N_1), Name(N_2), g_{N_1N_2}\rangle \in LSPD \\ 0 & \text{otherwise} \end{cases}$$

[2]We suppose that the name of a node N is that associated to the corresponding class.

4.1.3 Function η_m.

The function $\eta_m(N_1, N_2)$ takes in input two nodes $N_1 \in Net(S_1)$ and $N_2 \in Net(S_2)$ and returns a value denoting the similarity between $AN\_Set(N_1)$ and $AN\_Set(N_2)$. In order to carry out this task, we define a weighted bipartite graph $G(N_1, N_2) = (U \cup V, E)$; the plausibility coefficient computed for the pair (N_1, N_2) is obtained from a suitable objective function associated with the maximum weight matching computed over $G(N_1, N_2)$.

The computation of the objective function must take into account, for each pair of nodes $u_i \in U$, $v_j \in V$, both the similarity coefficients between u_i and v_j as well as the semantic relevance of u_i (resp., v_j) w.r.t. N_1 (resp., N_2).

The function η_m can be encoded as follows:

$$\eta_m(N_1, N_2) = \psi(LSPD, \rho(\tau(N_1, S_1), \tau(N_2, S_2)))$$

4.1.4 Function τ.

The function $\tau(N, S)$ takes in input a node N and the corresponding information source S and returns a set of pairs (N_i, r_i) such that $N_i \in AN\_Set(N)$ and r_i is the semantic relevance coefficient associated to the arc connecting N to N_i. The function can be formalized as:

$$\begin{aligned} \tau(N, S) = \{(N_i, r_i) \mid\ & N_i \in AN\_Set(N) \wedge \\ & \langle N, N_i, [d_i, r_i]\rangle \in A^{SDR}(S)\} \end{aligned}$$

4.1.5 Function ρ.

The function $\rho(PS_1, PS_2)$ takes in input two sets of pairs $PS_1 = \{(N_{1_1}, r_{1_1}), \ldots, (N_{1_p}, r_{1_p})\}$ and $PS_2 = \{(N_{2_1}, r_{2_1}), \ldots, (N_{2_q}, r_{2_q})\}$. ρ must modify either PS_1 or PS_2 in such a way that they have the same cardinality; this task is carried out by adding some fictitious pairs to the set having the lesser number of pairs. In particular, if $p < q$, ρ adds to PS_1 the set of $q-p$ fictitious pairs $\{(N_{1_{p+1}}, 0), \ldots, (N_{1_q}, 0)\}$; vice versa, if $q < p$, ρ adds to PS_2 the set of $p - q$ fictitious pairs $\{(N_{2_{q+1}}, 0), \ldots, (N_{2_p}, 0)\}$. Obviously names assigned to fictitious nodes are not significant so that no tuple of $LSPD$ can refer them.

4.1.6 Function ψ.

The function $\psi(T, PS_1, PS_2)$ returns a factor obtained from computing a suitable objective function of a maximum weight matching, as explained next. The input here are: *(i)* a set T of triplets denoting similarities between classes; *(ii)* two sets of pairs $PS_1 = \{(N_{1_1}, r_{1_1}), \ldots, (N_{1_p}, r_{1_p})\}$ and $PS_2 = \{(N_{2_1}, r_{2_1}), \ldots, (N_{2_p}, r_{2_p})\}$, as returned by the function ρ. The output is a value in the real interval $[0, 1]$.

First ψ constructs a suitable bipartite graph $BG = (U \cup V, E)$ where:

- U is a set of nodes; a node $u_i \in U$ corresponds to the node N_{1_i} of a pair $(N_{1_i}, r_{1_i}) \in PS_1$.
- V is a set of nodes; a node $v_j \in V$ corresponds to the node N_{2_j} of a pair $(N_{2_j}, r_{2_j}) \in PS_2$.
- E is a set of edges; in particular, there is an edge for each pair (u_i, v_j), $u_i \in U$ and $v_j \in V$. Each edge has a label $l_{ij} = [s_{ij}, r_{ij}]$; here $s_{ij} = f$ if $\langle Name(u_i), Name(v_j), f\rangle \in T$, 0 otherwise; $r_{ij} = 0.5 \times r_{1_i} + 0.5 \times r_{2_j}$.

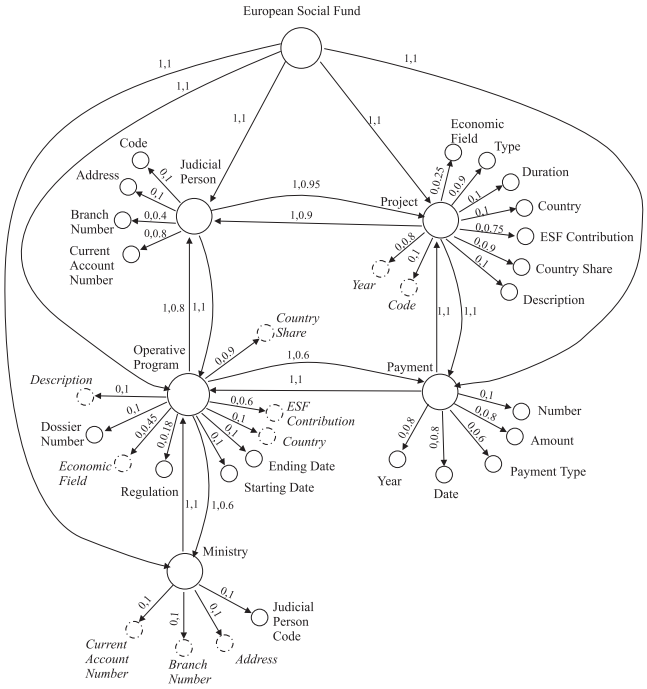

Figure 5: The SDR-Network of the ESF information source

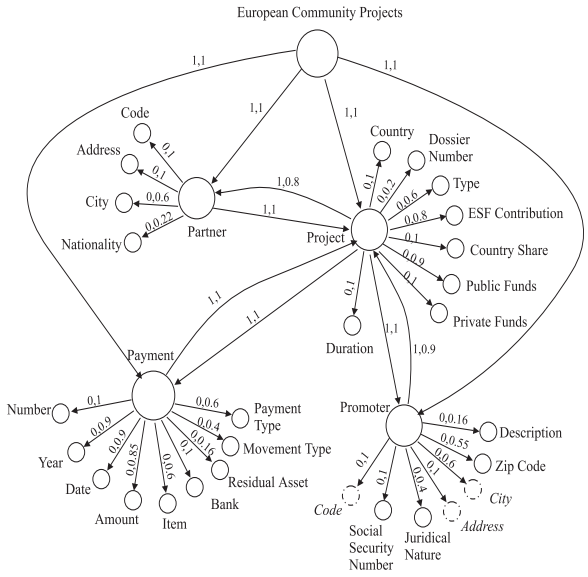

Figure 6: The SDR-Network of the ECP information source

The maximum weight matching for BG is a set $E' \subseteq E$ of edges such that for each node $x \in (U \cup V)$ there is exactly one edge of E' incident onto x and $\phi(E') = \frac{\sum_{\langle u_i, v_j, [s_{ij}, r_{ij}]\rangle \in E'} (s_{ij} \times r_{ij})}{\sum_{\langle u_i, v_j, [s_{ij}, r_{ij}]\rangle \in E'} r_{ij}}$ is maximum; we assume that if $\sum_{\langle u_i, v_j, [s_{ij}, r_{ij}]\rangle \in E'} r_{ij} = 0$ then $\phi(E') = 0$ (for algorithms solving maximum weight matching see [11]). The value returned by ψ, i.e. the objective function we have associated to the matching, is exactly $\phi(E')$.

The formula for $\phi(E')$ is based on the reasoning that, given two neighborhoods of two classes C_1 and C_2, the similitude of classes C_{1_i}, belonging to the neighborhood of C_1, and C_{2_j}, belonging to the neighborhood of C_2, contributes in the computation of the similitude of C_1 and C_2 in a way that depends on the semantic relevance of C_{1_i} w.r.t. C_1 and C_{2_j} w.r.t. C_2.

The example we present next should help reader's intuition about the behaviour of the presented technique.

4.1.7 A real example case

This example is derived from a series of tests we are carrying out on information sources of Italian Central Governmental Offices (ICGO). In this section we consider two of them, namely that storing information about European Social Funds (hereafter ESF) and that storing data on European Community Projects (hereafter ECP); the corresponding SDR-Networks are shown in Figures 5 and 6. In these figures a dotted node having label A is used to indicate that the arc incident onto A must be considered as incident onto the corresponding solid node having the same name. The adoption of this notation is motivated by layout reasons.

First we have constructed the $LSPD$ for class names belonging to ESF and ECP; it is shown in Table 1[3].

At the beginning BSD is constructed by calling $\eta()$; due to space limitations we show how $\eta()$ operates just on the pair of classes $Judicial\ Person$ of ESF and $Partner$ of ECP.

The similarity coefficient associated to this pair is:

$$g_{JudicialPerson,Partner} = \alpha_{lm}(JudicialPerson, Partner) \times \eta_l(JudicialPerson, Partner) + (1 - \alpha_{lm}(JudicialPerson, Partner)) \times \eta_m(JudicialPerson, Partner)$$

Since $\langle JudicialPerson, Partner, f\rangle \notin LSPD$ and $C\_AN\_Set(JudicialPerson, Partner)$ returns *true*, $\alpha_{lm}(JudicialPerson, Partner) = 0$. We must now compute $\eta_m(JudicialPerson, Partner)$.

$$\eta_m(JudicialPerson, Partner) = \psi(LSPD, \rho(\tau(JudicialPerson, ESF), \tau(Partner, ECP)))$$

Here:

$$\tau(JudicialPerson, ESF) = \{(Code, 1), (Address, 1), (BranchNumber, 0.4), (CurrentAccountNumber, 0.8)\}$$

$$\tau(Partner, ECP) = \{(Code, 1), (Address, 1), (City, 0.6), (Nationality, 0.22)\}$$

Since the cardinality of $\tau(JudicialPerson, ESF)$ is the same as that of $\tau(Partner, ECP)$, the function ρ must not add any fictitious node.

The function ψ first constructs a suitable bipartite graph and computes the maximum weight matching relative to it. In particular, selected edges are:

[3]We assume that if two classes A and B exist in ESF and ECP such that $Name(A) = Name(B) = N$, a triplet $\langle N, N, 1\rangle$ exists in the $LSPD$. Moreover, we assume that if the triplet $\langle C, D, f\rangle$ belongs to $LSPD$, also $\langle D, C, f\rangle$ belongs to $LSPD$.

Table 1: The *LSPD* associated to *ESF* and *ECP*

First Object	*Second Object*	Col_1		*First Object*	*Second Object*	Col_1
Ending Date	Date	0.8		Starting Date	Date	0.8
Number	Code	0.8		Branch Number	Bank	0.75
Country	Nationality	0.84				

$\langle Code, Code, [1,1]\rangle$, $\langle Address, Address, [1,1]\rangle$,
$\langle BranchNumber, City, [0,0.5]\rangle$,
$\langle CurrentAccountNumber, Nationality, [0,0.51]\rangle$.

Note that the last two edges of the matching have a weighing factor equal to 0 since the similarity between *BranchNumber* and *City* as well as that between *CurrentAccountNumber* and *Nationality* are 0; however, in the computation of $\phi(E')$, we take into account the corresponding relevance coefficient.

Now, $\phi(E') = \frac{1+1+0+0}{1+1+0.5+0.51} = 0.66$. As a consequence:

$g_{JudicialPerson,Partner} = \psi(LSPD, \rho(\tau(JudicialPerson, ESF), \tau(Partner, ECP)) = 0.66$.

Col_1 of Table 2 illustrates plausibility factors associated to some basic class similarities; in particular it shows all those whose participants are both complex nodes, plus some of those involving at least one atomic node.

4.2 Phase 2: Derivation of synonymies and homonymies

This phase takes the BSD derived by Phase 1 in input and detects synonymies and homonymies between classes of involved information sources.

Let S_1 and S_2 be two semi-structured information sources. Let C_1 (resp., C_2) be a class of S_1 (resp., S_2). We check the synonymy (resp., the homonymy) of C_1 and C_2 by computing a similarity factor f associated to C_1 and C_2 and by comparing it with a suitable, dynamically computed, threshold th_{Syn} (resp., th_{Hom}). If f is greater than th_{Syn} (resp., smaller than th_{Hom}) a synonymy (resp., an homonymy) is recognized for C_1 and C_2.

At the heart of our method for detecting similarity triplets $\langle C_1, C_2, f\rangle$, there is a fixpoint computation Γ^∞ over the SDR-Networks $Net(S_1)$ and $Net(S_2)$ associated to S_1 and S_2. The fixpoint computation starts with the base set of triplets stored in the BSD.

At the generic step i of the computation, the i_th neighborhoods of each pair of classes $C_1 \in S_1$ and $C_2 \in S_2$ are analyzed and their similarity coefficient is established using a maximum weight matching algorithm. This is exploited to refine the coefficient f associated with the similarity between C_1 and C_2 up to step $i-1$. The established similarity between $nbh(C_1, i)$ and $nbh(C_2, i)$ refines f in a way that is inversely proportional to i. Indeed, a decreasing succession $\{p(i)\}$ of factors is introduced to "weigh" the similarity factor obtained for $nbh(C_1, i)$ and $nbh(C_2, i)$ against the value of i.

The set of significant synonymies between classes belonging to involved information sources is computed by the function σ_{Syn} and is stored in the *Synonymy Dictionary SD*. σ_{Syn} takes in input a set of class similarities, as provided by the computation of the fixpoint of a suitable function Γ, and selects only those having a plausibility coefficient greater than th_{Syn}.

$SD = \sigma_{Syn}(\Gamma^\infty(BSD))$, where
$\sigma_{Syn}(T) = \{\langle C_1, C_2, f_{C_1C_2}\rangle \mid \langle C_1, C_2, f_{C_1C_2}\rangle \in T, C_1 \in S_1, C_2 \in S_2, f_{C_1C_2} > th_{Syn}(T)\}$

Here the threshold $th_{Syn}(T)$ is obtained as $th_{Syn}(T) = max\left(f_{max} \times \theta_{Syn}, th_m^{Syn}\right)$, where *(i)* f_{max} represents the maximum value of the plausibility factors associated to the tuples of T; *(ii)* θ_{Syn} is a tuning coefficient belonging to the real interval $[0,1]$; *(iii)* th_m^{Syn} is a minimum acceptable value for the plausibility similarity coefficient. We have experimentally set $\theta_{Syn} = 0.85$ and $th_m^{Syn} = 0.50$.

In an analogous way the *Homonymy Dictionary HD* (i.e., the set of significant homonymies between classes belonging to involved sources) can be defined:

$HD = \sigma_{Hom}(\Gamma^\infty(BSD))$, where
$\sigma_{Hom}(T) = \{\langle C_1, C_2, 1-f_{C_1C_2}\rangle \mid \langle C_1, C_2, f_{C_1C_2}\rangle \in T, Name(C_1) = Name(C_2), C_1 \in S_1, C_2 \in S_2, f_{C_1C_2} < th_{Hom}(T)\}$

Here the threshold $th_{Hom}(T)$ is obtained as $th_{Hom}(T) = min\left(f_{max} \times \theta_{Hom}, th_M^{Hom}\right)$, where the meaning of f_{max} and θ_{Hom} is as above, whereas th_M^{Hom} is a maximum acceptable value for similarity coefficients, which the corresponding homonymy coefficients are derived from. The optimal values for θ_{Hom} and th_M^{Hom} have been experimentally set to 0.33 and 0.27, resp.

4.2.1 *The fixpoint computation* Γ^∞.

By Γ^∞ we indicate a fixpoint computation; each step of the computation takes a triplet set T as its input and returns it modified by updating the similarity coefficients of the triplets of T. The fixpoint computation Γ^∞ is defined as follows:

$$\begin{cases} \Gamma^0(T) = T \\ \Gamma^i(T) = \Gamma(\Gamma^{i-1}(T), i-1) & for\ i > 0 \end{cases}$$

Now, for any information source S, let $K(S)$ be the minimum integer such that $K(S) > 0$ and $(\forall x \in S)(nbh(x, K(S)) = \emptyset)$. By Propositions 3.1 and 3.2, $K(S) \leq |A^{SDR}(S)| + 1$. Therefore we have:

PROPOSITION 4.1. *$\Gamma^\infty(T) = \Gamma^{\overline{K}}(T)$, where $\overline{K} = max(K(S_1), K(S_2))$*

PROOF. *Immediate.* □

The base functor Γ of the fixpoint computation Γ^∞ takes a set of triplets and an integer as its inputs and returns a set of triplets, as follows:

$\Gamma(T, i) = \{\langle C_1, C_2, \varphi_p(T, i, f, C_1, C_2)\rangle \mid \langle C_1, C_2, f\rangle \in T\}$

Table 2: Similarities derived between some classes belonging to ESF and ECP

First Object	*Second Object*	Col_1	Col_2		*First Object*	*Second Object*	Col_1	Col_2
Judicial Person	Partner	0.66	0.59		Judicial Person	Project	0	0.05
Judicial Person	Payment	0.28	0.30		Judicial Person	Promoter	0.51	0.47
Judicial Person	ECP	0	0.04		Project	ECP	0	0.02
Project	Partner	0.11	0.14		Project	Project	0.82	0.63
Project	Payment	0.15	0.18		Project	Promoter	0.10	0.14
Operative Program	Partner	0.10	0.13		Operative Program	Project	0.48	0.51
Operative Program	Payment	0.11	0.16		Operative Program	Promoter	0	0.02
Operative Program	ECP	0	0.04		Payment	ECP	0	0.07
Payment	Partner	0	0.03		Payment	Project	0	0.06
Payment	Payment	0.90	0.65		Payment	Promoter	0.23	0.25
Ministry	Partner	0.29	0.32		Ministry	Project	0	0.04
Ministry	Payment	0.14	0.17		Ministry	Promoter	0.26	0.22
Ministry	ECP	0	0.04		ESF	ECP	0	0.30
ESF	Partner	0	0.06		ESF	Project	0	0.03
ESF	Payment	0	0.04		ESF	Promoter	0	0.06
Address	Partner	0	0		Date	Project	0	0
Project	Country	0	0		Ministry	Year	0	0

4.2.2 Function φ_p.

Function $\varphi_p(T, i, f, C_1, C_2)$ returns the refined value of the plausibility coefficient for C_1 and C_2 and is defined as follows:

$$\varphi_p(T, i, f, C_1, C_2) = \begin{cases} p(i) \times \vartheta(T, C_1, C_2, i) + [1 - p(i)] \times f & if\ \vartheta(T, C_1, C_2, i) \neq 0 \\ f & otherwise \end{cases}$$

where:

- $\{p(i)\}$ is the succession of factors used to take into account distances between classes and their neighborhoods; the succession $\{p(i)\}$ is monotone decreasing since farther classes influence the similarity of C_1 and C_2 less than closer classes.

  Some interesting forms for $\{p(i)\}$ are the inverse polynomial $\left(p(i) = \frac{1}{(i+1)^k},\ k \geq 1\right)$ and the inverse exponential $\left(p(i) = \frac{1}{2^i}\right)$.

- ϑ measures the similarity between the $i\text{-}th$ neighborhood of C_1 and the $i\text{-}th$ neighborhhod of C_2.

4.2.3 Function ϑ.

$\vartheta(T, C_1, C_2, i)$ computes the similarity of $nbh(C_1, i)$ and $nbh(C_2, i)$. The similarity factor derives from an objective function associated to the maximum weight matching on a suitable bipartite graph obtained from the classes of $nbh(C_1, i)$ and $nbh(C_2, i)$. Therefore ϑ can be defined as follows:

$$\vartheta(T, C_1, C_2, i) = \psi(T, \rho(\tau'(C_1, i), \tau'(C_2, i)))$$

Functions ψ and ρ have been defined in Section 4.1.

4.2.4 Function τ'.

Function $\tau'(C, i)$ takes in input a class C and an integer i and returns a set of pairs (N_j, r_j) such that N_j is the target node of an arc belonging to $nbh(C, i)$ and r_j represents the semantic relevance of $ClassOf(N_j)$ w.r.t. C. In particular, the function τ' can be encoded as follows:

$$\tau'(C, i) = \{(N_j, r_j) \mid \langle N_k, N_j, [d_{kj}, r_{kj}]\rangle \in nbh(C, i),\ r_j = r_{kj} \times \varrho(\lfloor NodeOf(C), N_k \rfloor)\}$$

4.2.5 Function ϱ.

Function $\varrho(P)$ takes in input a path P and computes the PSR_P (see Definition 3), i.e., the Path Semantic Relevance of P.

4.2.6 A real example case (...continues)

We illustrate now the construction of the Synonymy Dictionary SD and of the Homonymy Dictionary HD relative to ESF and ECP. At first we must compute $\Gamma^\infty(BSD)$. In particular, we have:

$$\Gamma^0(BSD) = BSD;$$
$$\Gamma^1(BSD) = \Gamma(\Gamma^0(BSD), 0) = \Gamma(BSD, 0) =$$
$$\{C_1, C_2, \varphi_p(BSD, 0, f, C_1, C_2) \mid \langle C_1, C_2, f\rangle \in BSD\}$$

As for *Judicial Person* (hereafter JP) and *Partner* (hereafter Pa), the BSD stores the tuple $\langle JP, Pa, 0.66\rangle$ (see Section **4.1.7**). Therefore $\varphi_p(BSD, 0, 0.66, JP, Pa) = p(0) \times \vartheta(BSD, JP, Pa, 0) + [1 - p(0)] \times 0.66$. Here and in the following, we will adopt the quadratic decreasing function for $p(i)$, that is: $p(i) = \frac{1}{(i+1)^2}$.

Now, $\vartheta(BSD, JP, Pa, 0) = \psi(BSD, \rho(\tau'(JP, 0), \tau'(Pa, 0)))$.

$\tau'(JP, 0) = \{(Code, 1), (Address, 1), (BranchNumber, 0.4), (CurrentAccountNumber, 0.8)\}$

$\tau'(Pa, 0) = \{(Code, 1), (Address, 1), (City, 0.6), (Nationality, 0.22)\}$

Since the cardinality of $\tau'(JP, 0)$ is the same as that of $\tau'(Pa, 0)$, the function ρ must not add any fictitious node[4]. The function ψ first constructs a suitable bipartite graph and then computes the maximum weight matching relative

[4]Observe that the pairs given in input to ρ are the same pairs provided in input to the function ρ called during the computation of BSD (see Section **4.1.7**); it is worth pointing out that this is not a general rule.

to it. In particular, selected edges are $\langle Code, Code, [1,1]\rangle$, $\langle Address, Address, [1,1]\rangle$, $\langle Branch\ Number, City, [0,0.5]\rangle$, $\langle CurrentAccountNumber, Nationality, [0,0.51]\rangle$.

Now, $\phi(E') = \frac{1+1+0+0}{1+1+0.5+0.51} = 0.66$. As a consequence:

$\varphi_p(BSD, 0, 0.66, JP, Pa) = \vartheta(BSD, JP, Pa, 0) =$

$\psi(BSD, \rho(\tau'(JP, 0), \tau'(Pa, 0)) = 0.66$.

We continue with the fixpoint computation by considering $\Gamma^2(BSD) = \Gamma(\Gamma^1(BSD), 1)$. For the considered classes we have: $\varphi_p(\Gamma^1(BSD), 1, 0.66, JP, Pa) =$

$p(1) \times \vartheta(\Gamma^1(BSD), JP, Pa, 1) + [1 - p(1)] \times 0.66$.

Now, $\vartheta(\Gamma^1(BSD), JP, Pa, 1) =$

$\psi(\Gamma^1(BSD), \rho(\tau'(JP, 1), \tau'(Pa, 1)))$.

$\tau'(JP, 1) = \{(Project, 0.95), (Economic\ Field, 0.24),$
$(Type, 0.86), (Duration, 0.95),$
$(Country, 0.95), (ESF\ Contribution, 0.71),$
$(Country\ Share, 0.86), (Description, 0.95),$
$(Code, 0.95), (Year, 0.76), (Operative\ Program, 1),$
$(ESF\ Contribution, 0.6), (Country, 1),$
$(Ending\ Date, 1), (Starting\ Date, 1),$
$(Regulation, 0.18), (Economic\ Field, 0.45),$
$(Dossier\ Number, 1), (Description, 1),$
$(Country\ Share, 0.9)\}$

$\tau'(Pa, 1) = \{(Project, 1), (Country, 1), (Dossier\ Number, 0.2),$
$(Type, 0.6), (ESF\ Contribution, 0.8),$
$(Country\ Share, 1), (Public\ Funds, 0.9),$
$(Private\ Funds, 1), (Duration, 1)$

By calling functions ρ and ψ, we obtain $\vartheta(\Gamma^1(BSD), JP, Pa, 1) = 0.42$; therefore $\varphi_p(\Gamma^1(BSD), 1, 0.66, JP, Pa) = 0.25 \times 0.42 + 0.75 \times 0.66 = 0.60$.

Analogously $\Gamma^3(BSD) = \Gamma(\Gamma^2(BSD), 2)$. For the considered classes, the plausibility factor is

$\varphi_p(\Gamma^2(BSD), 2, 0.60, JP, Pa) =$

$p(2) \times \vartheta(\Gamma^2(BSD), JP, Pa, 2) + [1 - p(2)] \times 0.60 =$

$0.11 \times 0.60 + 0.89 \times 0.60 = 0.60$.

The computation of $\Gamma^4(BSD)$ causes a factor $\varphi_p(\Gamma^3(BSD), 3, 0.60, JP, Pa) = 0.06 \times 0.48 + 0.94 \times 0.60 = 0.59$ to be associated to considered classes. Since no other neighborhood exists for JP and Pa, we have that this is also the final value associated to the similarity between JP and Pa.

In the same way the plausibility values associated to all other entity pairs are computed; they are shown in the column Col_2 of Table 2.

Now, $SD = \sigma_{Syn}(\Gamma^\infty(BSD))$. Remember that:

$\sigma_{Syn}(T) = \{\langle C_1, C_2, f_{C_1C_2}\rangle \mid \langle C_1, C_2, f_{C_1C_2}\rangle \in T,$
$C_1 \in ESF, C_2 \in ECP, f_{C_1C_2} > th_{Syn}(T)\}$

In this case $th_{Syn}(\Gamma^\infty(BSD)) = max(0.65 \times 0.85, 0.50) = 0.55$. Therefore:

$SD = \{\langle Judicial\ Person, Partner, 0.59\rangle,$
$\langle Payment, Payment, 0.63\rangle,$
$\langle Project, Project, 0.63\rangle\}$.

An analogous reasoning leads to $HD = \emptyset$.

5. CONCLUSIONS

In this paper we have proposed a semi-automatic technique which aims at extracting synonymies and homonymies between object classes of heterogeneous, semi-structured information sources. In order to support the extraction task, we have introduced a new conceptual model and a related metrics for representing both object classes belonging to semi-structured information sources and their semantic relationships.

Presently, we are constructing a prototype implementing the presented algorithms and we are applying our technique to Italian Central Governmental Office information sources.

In the future we plan to extend the proposed methodology in various directions. In particular, we are working for *(i)* the development of algorithms for deriving other interesting interscheme properties, such as hyponymies, between object classes belonging to different semi-structured information sources; *(ii)* the definition of an integration algorithm which exploits derived interscheme properties for integrating semi-structured information sources.

6. REFERENCES

[1] S. Abiteboul. Querying semi-structured data. In *Proc. of International Conference on Database Theory (ICDT'97)*, pages 1–18, Delphi, Greece, 1997. Lecture Notes in Computer Science, Springer-Verlag.

[2] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener. The lorel query language for semistructured data. *International Journal on Digital Libraries*, 1(1):68–88, 1997.

[3] S. Abiteboul and V. Vianu. Queries and computation on the web. In *Proc. of International Conference on Database Theory (ICDT'97)*, pages 262–275, Delphi, Greece, 1997. Lecture Notes in Computer Science, Springer-Verlag.

[4] C. Batini and M. Lenzerini. A methodology for data schema integration in the entity relationship model. *IEEE Transactions on Software Engineering*, 10(6):650–664, 1984.

[5] C. Beeri and T. Milo. Schemas for integration and translation of structured and semi-structured data. In *Proc. of International Conference on Database Theory (ICDT'99)*, pages 296–313, Jerusalem, Israel, 1999. Lecture Notes in Computer Science, Springer Verlag.

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

[7] P. Buneman. Semistructured data. In *Proc. of Symposium on Principles of Database Systems, (PODS'97)*, pages 117–121, Tucson, Arizona, USA, 1997. ACM Press.

[8] P. Buneman, S. Davidson, M. Fernandez, and D. Suciu. Adding structure to unstructured data. In *Proc. of International Conference on Database Theory (ICDT'97)*, pages 336–350, Delphi, Greece, 1997. Lecture Notes in Computer Science, Springer-Verlag.

[9] S. Castano and V. De Antonellis. Semantic dictionary design for database interoperability. In *Proc. of International Conference on Data Engineering (ICDE'97)*, pages 43–54, Birmingham, United Kingdom, 1997. IEEE Computer Society.

[10] M. F. Fernandez, L. Popa, and D. Suciu. A structure-based approach to querying semi-structured data. In *Proc. of International Workshop on Database Programming Languages (DBLP'97)*, pages 136–159, Estes Park, Colorado, USA, 1997. Lecture Notes in Computer Science, Springer-Verlag.

[11] Z. Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*,

18:23–38, 1986.

[12] A. O. Mendelzon, G. A. Mihaila, and T. Milo. Querying the world wide web. In *Proc. of Conference on Parallel and Distributed Information Systems (PDIS'96)*, pages 80–91, Miami Beach (Florida), USA, 1996. IEEE Computer Society.

[13] S. Nestorov, S. Abiteboul, and R. Motwani. Inferring structure in semistructured data. *SIGMOD Record*, 26(4):39–43, 1997.

[14] S. Nestorov, S. Abiteboul, and R. Motwani. Extracting schema from semistructured data. In *Proc. of International Conference on Management of Data (SIGMOD'98)*, pages 295–306, Seattle, Washington, USA, 1998. ACM Press.

[15] S. Nestorov, J. Ullman, J. Wiener, and S. Chawathe. Representative objects: Concise representations of semistructured, hierarchial data. In *Proc. of International Conference on Data Engineering (ICDE'97)*, pages 79–90, Birmingham, United Kingdom, 1997. IEEE Computer Society.

[16] L. Palopoli, L. Pontieri, and D. Ursino. Automatic and semantic techniques for scheme integration and scheme abstraction. In *Proc. of International Conference on Database and Expert Systems Applications (DEXA'99)*, pages 511–520, Firenze, Italy, 1999. Lecture Notes in Computer Science, Springer-Verlag.

[17] L. Palopoli, D. Saccà, G. Terracina, and D. Ursino. A unified graph-based framework for deriving nominal interscheme properties, type conflicts and object cluster similarities. In *Proc. of Fourth IFCIS Conference on Cooperative Information Systems (CoopIS'99)*, pages 34–45, Edinburgh, United Kingdom, 1999. IEEE Computer Society.

[18] L. Palopoli, D. Saccà, and D. Ursino. Semi-automatic techniques for deriving interscheme properties from database schemes. *Data & Knowledge Engineering*, 30(4):239–273, 1999.

[19] L. Palopoli, G. Terracina, and D. Ursino. A conceptual model of information sources with heterogeneous structure for semantic represenation and derivation. Submitted for Publication. Available from the authors.

[20] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange across heterogeneous information sources. In *Proc. of International Conference on Data Engineering (ICDE'95)*, pages 251–260, Taipei, Taiwan, 1995. IEEE Computer Society.

[21] D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, and J. Widom. Querying semistructured heterogeneous information. In *Proc. of International Conference on Deductive and Object-Oriented Databases (DOOD'95)*, pages 319–344, Singapore, 1995. Lecture Notes in Computer Science, Springer-Verlag.

[22] D. Suciu. Semistructured data and xml. In *Proc. of International Conference on Foundations of Data Organization (FODO'98)*, Kobe, Japan, 1998.

[23] M. Tresch, N. Palmer, and A. Luniewski. Type classification of semi-structured documents. In *Proc. of International Conference on Very Large Databases (VLDB'95)*, pages 263–274, Zurich, Switzerland, 1995. Morgan Kaufmann.

[24] D. Ursino. Deriving type conflicts and object cluster similarities in database schemes by an automatic and semantic approach. In *Proc. of Symposium on Advances in Databases and Information Systems (ADBIS'99)*, pages 46–60, Maribor, Slovenia, 1999. Lecture Notes in Computer Science, Springer-Verlag.