Paper: jc11-7-2981; 2007/7/31

Coincidence-Based Scoring of Mappings in Ontology Alignment

Seyed H. Haeri (Hossein), Hassan Abolhassani, Vahed Qazvinian, and Babak Bagheri Hariri

Web Intelligence Laboratory, Computer Engineering Department, Sharif University of Technology and
School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (IPM)
E-mail: {shhaeri, abolhassani, qazvinian, hariri}@ce.sharif.edu

[Received January 31, 2007; accepted May 22, 2007]

Ontology Matching (OM) which targets finding a set of alignments across two ontologies, is a key enabler for the success of Semantic Web. In this paper, we introduce a new perspective on this problem. By interpreting ontologies as Typed Graphs embedded in a Metric Space, *coincidence* of the structures of the two ontologies is formulated. Having such a formulation, we define a mechanism to score mappings. This scoring can then be used to extract a good alignment among a number of candidates. To do this, this paper introduces three approaches: The first one, straightforward and capable of finding the optimum alignment, investigates all possible alignments, but its runtime complexity limits its use to small ontologies only. To overcome this shortcoming, we introduce a second solution as well which employs a Genetic Algorithm (GA) and shows a good effectiveness for some certain test collections. Based on approximative approaches, a third solution is also provided which, for the same purpose, measures random walks in each ontology versus the other.

Keywords: coincidence-based, ontology matching, metric spaces, genetic algorithms, graph theory

1. Introduction

In this section, first, an outline of the problem will be explained. A discussion on the terminology of this paper is given next. Afterward, a survey on the related works follows. This section is closed then by an outline of this work and its structure.

1.1. Outline of Problem

Semantic Web is said to be the next generation of Web where information is given a well-defined semantics in order to enable computer agents to use them in the same way that human beings do. Unlike traditional knowledge-based systems, as like as the web itself, Semantic Web is **by design** distributed and heterogeneous. Ontologies are aimed to play a central role in making this heterogeneity feasible while simultaneously making it possible to reason about this distributed knowledge. However, in many real cases, since they are created by diverse parties distant from each other (and possibly with a very little shared knowledge), the ontologies themselves also suffer from heterogeneity. The need thus arises for a mechanism to tackle this heterogeneity to enable computer agents to leverage the semantic interrelationships among the entities of ontologies during reasoning processes. The set of mechanisms for dealing with this is usually referred to as Ontology Alignment (OA), where Ref. [1] defines it as:

> ... given two ontologies which describe each a set of discrete entities (which can be classes, properties, rules, predicates, etc.), find the relationships (e.g., equivalence or subsumption) holding between these entities.

OM, meanwhile seems to be a subtask of OA. A lot of current OM methods calculate inter-conceptual similarities using some predefined measures (phase 1), and via the interpretation of results, put forward some possible set of semantic interrelationships among the entities (phase 2). Given O_1 and O_2 as the ontologies we are to align, and defining $O = O_1 \cup O_2$, typically a *dissimilarity* (or *distance*) measure is formally defined as follows [2]:

A dissimilarity $\delta : O \times O \to \mathbb{R}$ is a mapping from a pair of entity to a real number – expressing the distance between two objects such that:

$\forall x, y \in O, \delta(x,y) \geq 0$ (*positiveness*)
$\forall x \in O, \delta(x,x) = 0$ (*minimality*)
$\forall x, y \in O, \delta(x,y) = \delta(y,x)$ (*symmetry*)

It is customary to have **dis**-similarity in the scale of 0 to 1 to define similarity as "1 − dissimilarity".

The many different similarity measures defined in the literature are generally categorized into two groups: *lexical* and *structural*. Lexical measures are concerned about lexicographical similarity, while structural measures leverage hierarchical relationships among concepts (e.g., number of common children, common parents, etc.).

It is common also to first define a set of similarity measures – lexical or structural – then apply them consecutively like a *compound similarity measure* (**Fig. 1**). The application of this set of (compound) similarities yields an initial guess. The final decision is made afterward in another phase. In this phase, the ultimate set of satisfactory correspondences between the ontologies is defined. In this view, mapping extraction is a process to find the best mapping across ontologies.

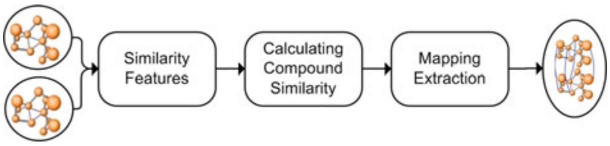

Fig. 1. A simplified alignment framework.

1.2. Terminology

It is worth mentioning that we suspect, without this subsection, some precise readers may feel confused about how we use the terms OA and OM. This clarifier subsection, is contrived for removal of such confusions.

Firstly, to our knowledge, there exists no consensus on a precise definition of these two terms. We therefore adopt the following definitions, which appear to be well-respected in the literature:

Reference [3] defines the term *Mapping* as:

> a formal expression that states the semantic relation between two entities belonging to different ontologies. When this relation is oriented, this corresponds to a restriction of the usual mathematical meaning of mapping: a function (whose domain is a singleton[1]).

And, then, defines OA accordingly as:

> a set of correspondences between two or more (in case of multi-alignment) ontologies (by analogy with DNA sequence alignment). These correspondences are expressed as mappings.

The OA definition quoted in Section 1 from [1] appears to agree with this. Additionally, OM is defined in [4] as:

> the problem of finding the semantic mappings between two given ontologies.

(Note that Ref. [4] is speaking about "finding *the* semantic mappings" rather than *a set of* semantic ones. This suggests that Ref. [4] assumes the existence of some particular mapping that is superior to any alternative.)

Putting all these definitions together, our understanding is that OM is **the act** of finding **some proper *alignment*s** each of which, in return, is a set of mappings. This is how hereafter we apply our terminology in this paper, which also agrees with [5]. Note, however, that we do not assert the existence of any consensus on the working definition chosen for this paper. Alongside, the reader might also note that the CFP of ESWC 2007 for example, includes [6]:

> Topics of interest to the conference include (but are not restricted to):
>
> ...
>
> - Ontology Alignment (mapping, matching, merging, mediation and reconciliation)

which implies that OA is a set of tasks, one of which is OM.

A final remark which, theoretically speaking, is much more important is that the understanding of the Semantic Web community of the term "matching" apparently clashes with that of Mathematics. Given that our work – like a bewildering number of other related ones – is greatly engaged with Mathematics, let those readers coming from a mathematical background be warned that they will probably become confused if they adhere solely to their prior terminology.

1.3. Related Works

Current researches in ontology mapping and its applications entails a large number of fields ranging from machine learning, concept lattices, and formal theories to heuristics, and linguistics. Similar attempts have also been done to match graphs, and trees [7, 8], database schema [9] and even in clustering compound objects with a machine learning technique [10]. Yet, works on ontologies and mapping extractions are not so many [3].

Although there are works which choose to address both simultaneously, the works related to that of ours generally choose to work on either:

- alignment weighting and similarity measures; This group of works mainly focuses on the similarity measures (across the concepts of the two ontologies) and weight functions. The purpose is to evaluate a given alignment. Or,
- mapping extractions, in which the research tries to address extracting alignments and proposing methods to find a (more proper) alignment.

We will have a quick review on each category in the two following subsections.

1.3.1. Alignment Weighting and Similarity Measures

Some standards of metrics are acknowledged and defined as in the CommonKADS methodology [11], or OntoWeb EU thematic network [12], which are partly endorsed by recognized bodies. Also, there have been some works on finding similarities of entities in two ontologies based on their structural standings: Ref. [13] computes the dissimilarity of elements in a hierarchy based on their distance from closest common parent. The Upward Cotopic distance is introduced by [14] where they found dissimilarity of entities in hierarchies of ontologies. The key difference between those works and the current one is that they consider the mere structural features of ontologies.

Reference [15] introduces a measure to calculate similarity of WordNet[2] concepts, i.e. a single hierarchy. The similarity is computed based on the closest common parent and distance of the two entities from the root. This work gets closer to that of ours but it is very immature in that it very simply presumes a hierarchical structure for

[1] Unfortunately, mathematically speaking, this is incorrect because no such constraint exists on functions in mathematics.

[2] wordnet.princeton.edu

every ontology. The authors of this paper understand that this is an engineering assumption. Yet, we believe that this is far away from correct in reality and that our work has no such assumptions.

On the other hand, some methods tend toward a tradeoff between different features such as efficiency and quality, as in QOM [16], and some have used approaches to integrate various similarity methods [17]. This work, unlike them, offers a manifesto of its desired properties. Then, it examines a few solutions which adhere to that.

Besides, compound metrics get use of simple measures by combining them, and hoping to improve the result of the mapping between two ontologies. One approach has been to define each measure as a dimension to find the Minkowski distance of two objects [18]. As introduced in [18], another approach for this problem has been weighted average of features in which weight can even be defined by a machine learning technique. Glue [19] builds the similarity matrix by a machine learning approach too. In APFEL [20] weights for each feature are calculated using Decision Trees. The user only has to provide some ontologies with known correct alignments. The learned decision tree is then used for aggregation and interpretation of the similarities. Ref. [21] introduces a new method for compound measure creation without any need to the mapping extraction phase. It estimates the similarity among entities of two ontologies based on existing transitive relationships across the ontologies.

1.3.2. Mapping Extraction

A method for mapping extraction is proposed by [22] which examines linguistical features to compare two ontologies on the basis of an *IS-A* relationship. Staab et al. [23] have also focused on structural and taxonomic comparison of two trees. To extract an alignment, dissimilarity of each two concept is calculated based on their superclasses and subclasses. Stumme et al. [24] uses shared instances of two ontologies that are to be mapped, however this work ignores the properties of classes. Again the preference of our work over these ones is that it is not biased towards any special way in which the ontology (as a graph) is shaped or how the labels are used.

Zhdanova et al. [25] expand the notion of OM to a community-driven approach to enable web communities to establish and reuse OM to achieve an adequate and timely domain representation. Our work in contrast is not targeting any special domain.

In [26], to extract a reasonable alignment, applicability of the solutions for the *Stable Marriage* [27] problem is studied. There are some other approaches, as an example, a machine learning approach to the problem is discussed in [4], and Ref. [28] describe a probabilistic-based model.

Johnson et al. [29] model inter-ontology relationship detection as an information retrieval task, where relationship is defined as any direct or indirect association between two ontological concepts.

Wang et al. [30] presents a specific formalization and algorithm presented for local interpretation of shared representations to build global semantic coherence for the distributed actions of individual agents, known as *Mutual Online Ontology Alignment*.

LOM as described in [31] is a semi-automatic lexicon-based ontology-mapping tool that supports a human mapping engineer with a first-cut comparison of ontological terms between the ontologies to be mapped (based on their lexical similarity and simple heuristic methods). These works, unlike that of ours, are mostly careless about the (overall) structure of the ontology.

1.4. This Work

This paper introduces a new factor called *coincidence* that combines different ideas from different realms of science and engineering, including Ontology Matching, Graph Homeomorphism, Metric Spaces, and Domain Theory. In simple words, it targets scoring the mappings based on how graphically better the coincidence of ontologies appears for different mappings. Therefore, it can be used in phase 2 of an alignment framework. This work enumerates the properties which a measure with such a quality should have, and offers one such measure itself. Then, to demonstrate this use in action, it gives three approaches for mapping extraction based on this measure.

In the simplest form, we generate all possible alignments and score each based on the measure, and finally, select the ones having maximum scores (global maximals). However, this method suffers from exponential runtime and, therefore, has a limited application (to small ontologies). For attaining a more docile solution for large ontologies and generating a nearly optimal solution, we developed a Genetic-Based algorithm which applies the coincidence measure during generation of new individuals such that new generations have better coincidence. We also developed an approximative approach which does not insist on generating all the alignments first and then estimating their scores. Instead, it attempts to **estimate** the mapping having the best coincidence score.

To introduce the coincidence measure, the basic mathematical background is explained in Section 2.1 and the corresponding problem is defined formally in Section 2.2. In Section 3, we introduce the measure in Section 3.1 by discussing the intuitions that the solution is based upon. Translation of the intuitions into different possible graph structures comes in section 3.2. The formulation of a scoring mechanism is explored in Section 3.3, in addition to some commentary on the mechanism in Section 3.4. Moreover, in Section 4, we show how to use the mechanism in three different ways by first having a discussion on how to reduce complexity for OWL ontologies in Section 4.1. Explanation of a naive approach is in Section 4.2, a Genetic Based approach in Section 4.3, and an approximative one in 4.4. Finally, we present a conclusion in Section 5.

2. Specification of the Problem

2.1. Mathematical Background

In this section, we define necessary mathematical concepts which are used throughout the paper. The first one is the notion of a *Metric Space* for which we refer to what defined in [32]:

> A set X, whose elements we shall call points, is said to be a metric space if with any two points p and q in X there is associated a real number $d(p,q)$, called the distance from p to q, such that:
>
> $d(p,q) > 0$ if $p \neq q; d(p,p) = 0;$ [*positiveness*]
>
> $d(p,q) = d(q,p);$ [*symmetry*]
>
> $d(p,q) \leq d(p,r) + d(r,q), \forall r \in X.$ [*triangular inequality*]
>
> Any function with these three properties is called a distance function, or metric.

Another piece of theory which is of help is the notion of *Typed Graphs*[3]. In general, we call a graph G *typed* if each edge of it has a type. In other words, let us formally define $G(V,E,T)$ a typed graph if $E : V \times V \to T$, where T is a set of predefined types. An edge e of type t is written $e : t$. A homeomorphism from a typed graph $G(V,E,T)$ to a typed graph $G'(V',E',T)$ is a one-to-one correspondence m between V and V'. We will call an edge $e(a,b) : t \in E$ *preserved under* m or $P(e,m)$ iff there is an edge $e'(m(a),m(b)) : t \in E'$. If both a and b get mapped to some vertex in V', yet there is no edge of type t between $m(a)$ and $m(b)$ – *typelessly preserved*, we write $TP(e,m)$. We will call a typed graph $G(V,E,T)$ vertices of which are points in (X,d) *embedded in* X, and write $G(V,E,T,X,d)$.

Reference [33] defines a *Partially Ordered Set* as follows:

> A set P with a binary relation $\sqsubseteq$ is called *a partially ordered set* or *poset* if the following holds for all $x,y,z \in P$:
>
> 1. $x \sqsubseteq x$ (Reflexivity)
> 2. $x \sqsubseteq y \wedge y \sqsubseteq z \Rightarrow x \sqsubseteq z$ (Transitivity)
> 3. $x \sqsubseteq y \wedge y \sqsubseteq x \Rightarrow x = y$ (Antisymmetry)

We add that $\sqsubseteq$ above is called a *partial order*.

As the last definition, let us call the set of all the directed paths stemming from a vertex v the set of *v-stems*. A path in this set will, analogously, be called a *v-stem*. It should be noted that an implication of this definition is that v should be in a directed graph basically to have a stem.

2.2. Theoretical Specification of Scoring

Assume that we are given two Ontologies as well as distance values for each pair of concepts across the ontologies. Such distances may have been obtained by application of a (lexical, structural, or compound) measure. The goal is to score mappings (and thereafter alignments) so that one can select a best or near-the-best alignment among all the available possibilities. We formulate this problem as follows:

Input: A pair of ontologies, and a matrix, rows and columns of which stand for concepts from one ontology, and concepts from the other, respectively. Each cell shows the distance between the corresponding concepts.

From our point of view, this input is interpreted as a pair of directed acyclic graphs embedded in a metric space. So, naming the input ontologies O and O', we do not distinguish them from $G(V,E,T,X,d)$ and $G'(V',E',T,X,d)$ respectively.

Output: A scoring of possible alignments which can be a help for better extraction. From our point of view, this is a partial order on the possible homeomorphisms between G and G'.

To produce the above output, this paper first enumerates a list of rationales for the above partial order, and then presents one possible candidate for that. This leads to a straightforward yet non-effective solution. We will then discuss possible axes along which one can tune that and add two related solutions that overcome the complexity of the first one.

We should mention that – although some experts may consider our work a method for mapping extraction – we believe that this part offers a new criteria which helps **deciding better on extraction**, as opposed to extraction itself.

3. The Partial Order

In this section, we first give an intuition for our method, translate that intuition to various graph patterns and finally give a precise specification of our scoring mechanism.

3.1. Intuition

We forget about OA for a few minutes, and consider the following basic-geometry problem: When do we call a pair of triangles **the same**? When they are equal in the geometric sense? For example, do we consider the two triangles in **Fig. 2a the same**? We do doubt[4]! Now, looking at **Fig. 2b**; (The solid lines indicate one triangle, the dotted ones indicate another, while the vertices of the triangles coincide.) Up to our understanding, we – human-beings – consider these two triangles **the same**! Now, introducing the case of **Figs. 2c** and **2d** into the comparison, and trying to give a fuzzy interpretation to the concept of "being the same" – or **coincidence**, it should be said that: the two triangles in **Fig. 2d** are more the same than that of **Fig. 2c**. And, the two of **Fig. 2b** coincide even more.

Back to the realm of OA, the authors should say that the approaches which are concerned merely about the structure of ontologies are imprecise in that they fail to distinguish between the two triangles in **Fig. 2a**. That is to say,

[3] There is no consensus in mathematics on this name.

[4] In mathematical topology, these two triangles are the same in that there exists a continuous bijection between them, inverse of which is also continuous.

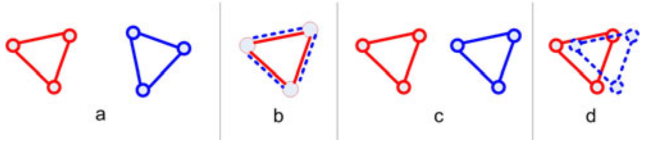

Fig. 2. Matching of shapes.

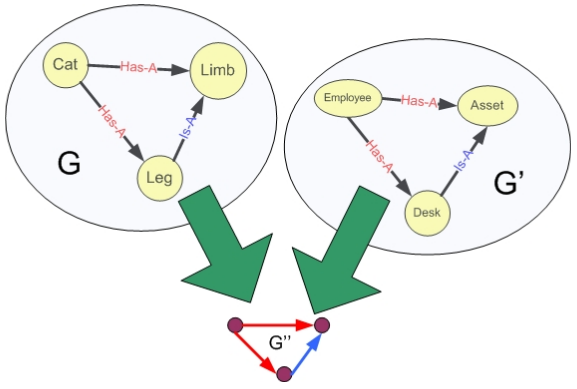

Fig. 3. Structure is not enough.

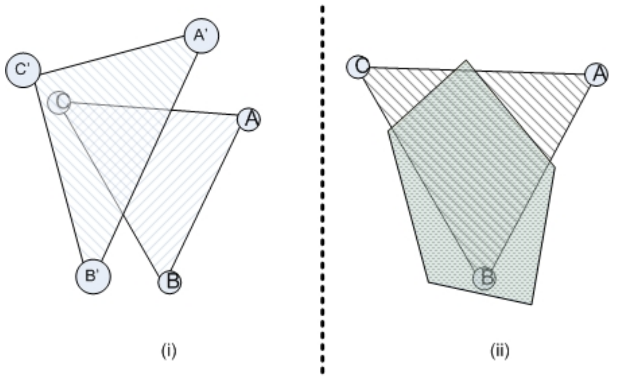

Fig. 4. The impact of the correct choice for mapping.

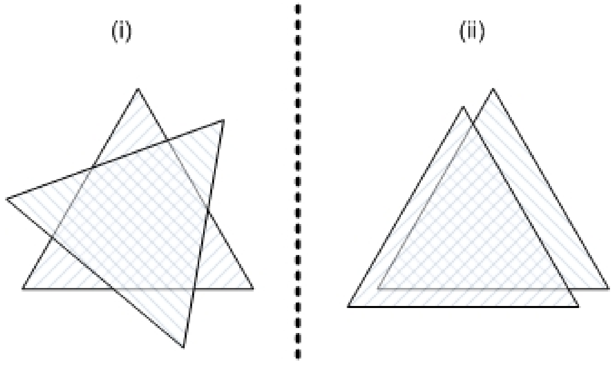

Fig. 5. Coincidence is not only being close.

as **Fig. 3** depicts, those approaches tend to reduce both G and G' to the same graph (G''). This, obviously, is a very magnificent loss of information because G and G' will be interpreted as similar ontologies while they are describing totally different worlds.

For the next step to understand the notion of coincidence and their usage in OA, we consider **Fig. 4** where all the points are considered to be in a metric space. Suppose that we are about to have an estimate for *how much* the two triangles of part (i), namely ABC and $A'B'C'$, coincide. One may find it trivial that this is a function of $d(A,A')+d(B,B')+d(C,C')$, where d is the metric of our metric space. What this means is that we tend automatically to choose A to be paired to A', B to B', and C to C'. The reason why this happens is that this way, by merely pairing each vertex to its closest counterpart from the other triangle, the overall distance of the two triangles will be minimized too. That is to say, naturally, human-beings do not try to estimate the distance between the two triangles by considering $d(\mathbf{A, B'})+d(\mathbf{B, A'})+d(C,C')$. Because this latter sum will needlessly be more than the former.

Considering the same problem for the triangle and pentagon in **Fig. 4(ii)** will not be this trivial. One has to be careful about how to pair the vertices up so that the overall sum minimizes. This is the case because each choice affects the rest of vertices too. The problem will become more sever when one is dealing with complicated shapes with large number of vertices. This is where the matter of how to pair up the vertices – i.e., mapping and alignment – becomes a keynote. One can observe now that different alignments can affect the way coincidence of ontologies get interpreted. For that, Section 3.2 lists the properties that are expected from a good interpretation from the degree of coincidence of two graphs. As it turns out, those properties are dependent on the mappings and will therefore help us to identify the alignments which helps us to have a better understanding over the coincidence of two ontologies.

This is what we are about to inject in the world of OA. That is, given that the phase one of OA gives us a measure for similarity of concepts across the ontologies, we consider this measure as an estimate for the distance between each pair of points (i.e., concepts), and suite it for estimating the extent to which the two ontologies – as the whole graphs – **coincide**. Alongside, we first offer an estimate for the extent of coincidence between two edges, and then accumulate all these as our final estimate for the coincidence of the two ontologies.

It is worth mentioning that it may be tempting to forget about the differences between Metric and Cartesian Spaces, and mistakenly think about coincidence as merely being close. With that misconception, one might decide to define coincidence in terms of a centroid. Regardless of the technical difficulties that defining centroid in a Metric Space has, we should mention that this approach will not describe coincidence. In **Fig. 5** for example although the centroid of the two triangles in part (i) exactly coincide, the two triangles themselves do not. A comparison between this part and part (ii), will reveal it that in spite of the fact that there is a distance between the centroids of the latter pair of triangles, they happen to be more coinciding. This observation tells us that coincidence is rather a direct function of all the pairwise distances of the nodes than a single representative (such as centroid).

The astute reader may wonder what technical difficulties might defining centroid for ontologies have. Here is an interesting one: First we should mention that a centroid is usually defined to be the point which has equal distance from all the points of a shape. Then, what is the interpretation of such a point for ontologies, if any? Furthermore, assuming that, for every ontology, we can find a proper point in our Metric Space with such a property, it carries no significantly meaningful information for the other ontologies. For example, if $O \subset O'$ and the majority of the concepts of O' are far away from O, so will be the centroid. In this case, considering O and O' to be non-coinciding is an obvious mistake – yet the distance between the centroids will be significant. The technical difficulties of centroids in Metric Spaces is not limited to OA. For instance, one can refer to [34] for an excessive list of such difficulties in capturing proximity of webpage elements.

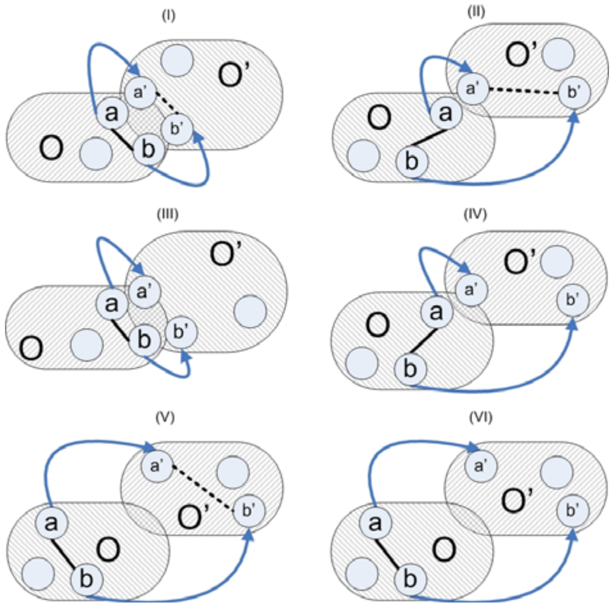

Fig. 6. Properties of metrics.

3.2. Properties of the Desired Partial Order

Here will be a set of properties which we believe any partial order for our problem should convince, along with our reasons for such beliefs. Our proposed partial order is in fact a *weight function* for matchings, so hereafter we use *weight* in place of it. The set of properties are divided into 6 categories, based upon preservation of the edge (under the correspondence), and upon the mutual distance between its heads.

In all categories of **Fig. 6**, O and O' are the input ontologies, a and b will be concepts in O, and, a' and b' concepts in O'. The closer a pair of concepts is depicted in figures, the closer the concepts are intended to be in the (X,d). (That is, the closer a and a' are shown in the figures, the smaller $d(a,a')$ is.) We do not force the ontologies to be disjoint, so, in each figure, it can be seen that the surface of ontologies may overlap. Furthermore, in each figure, the arrows show mappings. (That is, the source of arrow is intended to be said is mapped to its destination.) And, the lines – be it solid or dotted – show the edges of graphs. (Solid lines show the edges between a and b, and dotted edges show the edges between a' and b'.)

Category I. Here, a and a' are too close, like b and b'. The fact that (a,b) is preserved is of much importance to us because it means that the two **edges** coincide too much. So, we want this preserved edge to bring a great weight. To justify it, consider the case when a and b are "Animal" and "Jaguar" respectively, and a' and b' are "Living Creature" and "Tiger". The fact that there is an edge (of type *redfs:type*) between both a and b, a' and b', means very much that the two ontologies are perhaps describing the same world.

Category II. In this category, the edge is preserved, but only one peer of the edge is close to its image. As an example of such cases, one can consider O be describing a Zoo, and O' a Museum. Furthermore, suppose that a and b are "Elephant" and "4-legged", and, a' and b' are "Mammoth" and "Ancient Creature". An interpretation of this is that although O and O' are describing two different worlds, they are perhaps getting coincident "from the side of a". Therefore, we would like such cases to get a moderate weight, i.e., smaller than the previous case.

Category III. The third category is the one where an edge is not preserved while the relevant concepts are so close. Consider, e.g., when O is describing the Glazing Technology, while O' is the ontology of a simple glasses manufacturing studio. In this respect, a and b could be "Glass" and "Frame", and a' and b' the same respectively. Of course $d(a,a')$ and $d(b,b')$ may both be very small here. We consider the non-preservation of edge a negative point, but because the vertices coincide, we do not penalize this matching that much. This is logical because the closeness of (a,a') and (b,b') means that the edge (a',b') is perhaps mistakenly missed.

Category IV. Next, we come to the category where an edge is not preserved, while only one side of the edge is too close to what it is mapped to. A mapping which does this is perhaps trying to make a mistake, but not as big as category VI. So, we will not penalize it that much. As an example of such a case, we consider this case: O is describing a glasses manufacturing studio, and O' is a car factory. Assume that a is "Glasses" and a' is "Glass", b could be "Frame", while b' is "Chassis". Like category III which is somehow dual of category I, this category can be considered dual of category II.

Category V and VI. A preserved edge certainly increases the likelihood of preservation of shape for the two entire graphs. However, if neither endpoint of the edges are close to what they are mapped to, the two edges do not coincide that much. This does not to be a great success, therefore, because it does not greatly help the coincidence of two ontologies. In other words, although the preserva-

Table 1. The six categories and their treatments.

Proximity⇒ ⇓Type of Edge	**Both** Ends Close	**One** End Close	**Neither** End Close
Preserved	High Benefit	Modest Benefit	Low Benefit
Not Preserved	Low Penalty	Modest Penalty	High Penalty

tion of shape (as depicted in **Fig. 2a**) is partly important, we do not care that much about it if the edges coincide at neither end. For an example of when this looks rational, we consider the case when a is "Vehicle", b is "4-wheeled", a' is "Animal", and b' is "4-legged". Therefore, for the category V, we would like the mapping to receive a low benefit. The situation is completely similar to that of category VI, so, we do not try to justify why a mappings of that category will be penalized to a large extent.

Table 1 summerises the above manifesto about the six categories along with our suggested treatment for each case.

3.3. Our Proposed Partial Order

Adding the fact that the weighting system is expected to be symmetric in its arguments, we observed that one possible such weighting is the following[5] in which $v_1, v_2 \in G$: (By being symmetric in its argument, we mean $w(m(G,G')) = w(m^{-1}(G',G))$.)

$$w(m) = w_0(m) - w_l(m) - w_r(m),$$

where

$$w_0(m) = \sum_{P((v_1,v_2),m)} \overline{f}_m(v_1) + \overline{f}_m(v_2)$$

$$w_l(m) = \sum_{TP((v_1,v_2),m)} \overline{g}_m(v_1) + \overline{g}_m(v_2)$$

$$w_r(m) = \sum_{TP((m(v_1),m(v_2)),m^{-1})} \overline{g}_m(v_1) + \overline{g}_m(v_2)$$

$$\overline{f}_m(x) = 1/f(d(x,m(x)))$$

$$\overline{g}_m(x) = g(d(x,m(x)))$$

The interpretations for w_0, w_l, and w_r are:

- w_0: This part of formula is in charge of accumulating the score of coincidence of all the preserved edges (under m).
- w_r: This part accumulates the coincidence which is *missed* because of typeless preservation of edges of G under m. And, finally
- w_l: is the accumulated loss of coincidence because of typeless preservation of edges of G' under m^{-1}.

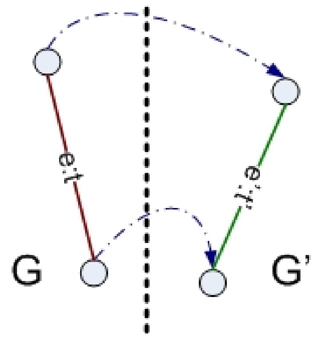

Fig. 7. w_r and w_l are not the same.

Although w is symmetric in its arguments, w_r and w_l are not the same. **Fig. 7** demonstrates an example where $t \neq t'$, and the typeless preservation of the edge $e : t \in G$ gets counted only in w_l (and not in w_r). Likewise, the typeless preservation of the edge $e' : t' \in G'$ is only counted in w_r. As a result, the respective mapping receives two penalties for these two edges; one for e and another for e'.

We do not claim any validity for the functions f and g; because they are meant to be experimentally tuned. That is to say, these functions can be considered as **normalization** functions. Their common property is being strictly increasing. Otherwise, one can always find one of the six categories above in which w will misbehave. Furthermore, f should have another property as well; its range should be outside a certain neighborhood around origin. For the case when this will result in a misbehavior, consider a pair of ontologies across which there exists a pair of concepts with distance 0. If $f(0)$ is 0, then w will become $+\infty$, regardless of the rest of alignment. And, this obviously is a significant anomaly because it will cause a big class of alignments to look the same – while they are not inherently the same. That is, in such a case, w does not do much for a large class of alignments.

In presence of a vertex which does not get mapped to anything, all the edges from that vertex – or to it – are not preserved. In these cases, the alignment should get more weight than one which has mapped such edges to edges with wrong types. To tune our formula to reflect this, virtually consider it being mapped to an imaginary vertex, existence of which does not give us any information. In this case, its distance ought to be 0 from any other concept. One can easily verify it that the above weight satisfies all the conditions enumerated. As a further benefit of our proposed weighting method, we would like to notify it further that, for cases like that of **Fig. 7**, our weighting method would penalize m twice; once because e is not preserved, and another time for e'.

The special case where this will become more interesting even is when e : *subClassOf*, and e' : *subClassOf*. Here, our weight will recognize the fact that an alignment which maps e endpoints to two concepts between which there is no edge at all, is better than when they get mapped to a pair of edges where there is an edge between them **with an inverse type**.

[5] For a note on how to prevent this formula to approach to infinity, please refer to Section 3.4.

3.4. Commentary

As said before, there are cases in which what the input matrix gives us may not be a metric space. In fact, as said in Section 2.1, a metric space is needed to have symmetry. However, as listed in [5], there are schema-based matching techniques which use linguistics resources. These techniques may not convince this property. That is, for example: In the Webster Collegiate Dictionary [35], "quick" is in the 12^{th} place in the list of synonyms of "swift", while "swift" is second in the list of synonyms of "quick". In such a case, the symmetry property may not hold. Therefore, what we get may be a *Quasi-Metric Space* [36] rather than a metric space. However, as [3] also mentions, only few authors may consider similarity metrics which do not have symmetry. So, the existing weighting formula and the assumption with it will almost always be convincing. Even in case where one is faced with an application in which there inherently exists no symmetry, a little tweak to the formula will give rise to a **symmetric weighting formula** which still convinces all the conditions listed in section 3.3:

$$w'(m) = w_0(m) - w_l(m) - w'_r(m)$$

where $w_0(m)$ and $w_l(m)$ remain the same, but

$$w'_r(m) = \sum_{TP((m(v_1),m(v_2)),m^{-1})} \overline{g}_{m^{-1}}(m(v_1)) + \overline{g}_{m^{-1}}(m(v_2))$$

Furthermore, there seems no way to guarantee that the triangular inequality holds for **any** output of the phase 1. Despite that, it seems quite reasonable to assume that this property holds for any such guess. In fact, we believe finding a **real** guess in which this does not hold is unlikely.

Another question which may arise is about complexity. It can easily be shown that naively using these formulas needs an exhaustive search; finding the best mapping directly is not known to be $\mathscr{P}$ or $\mathscr{NP}$. Suppose on the contrary that it is efficient. Then, one can come to an efficient way for solving the graph isomorphism problem; given a pair of (un-typed) graphs (not embedded in a metric space), assign a fixed type t to all of the edges, embed them in a metric space in which the distance of any pair of points is 1, and run our algorithm on them – **in an efficient time**. The heaviest matchings can be efficiently checked for being an isomorphism, because one can remove the types and the metric space backbone. It is easy to verify that there is a homeomorphism between the original graphs iff the correspondence with the biggest weight is a isomorphism between them. This will give us an efficient way of solving the graph isomorphism problem. This means that we now know that this latter problem is $\mathscr{P}$ – which of course we do not.

So far, we assume that for considering all the possible matchings, one iterates through alignments until making sure that they are finished. This means the algorithm iterates exponential times. Nevertheless, considering all the possible matchings is not needed. As Papadimitriou and Steigiltz show in [8], there exist heuristics for dealing with this in a $\mathscr{P}$ time. For the moment, however, we do not consider those heuristics. Despite that, we are not about to leave this problem in its general form; We believe that the OM-specific heuristics presented 4.1 can decrease the runtime. However, for fully supporting the exponential nature of exhaustive search, one needs more elaborated approaches such as the ones offered in sections 4.3 and 4.4.

4. Alignment Selection

In this section, we explain about three possible ways to use the explained solution for the alignment selection (also referred to as mapping extraction). First, we explain about some heuristics for decreasing the runtime. Based on that, a trivial approach is explained. Next, trying to come up with more elaborated solutions, a GA approach is introduced. This section is then finalized by an approximative approach.

4.1. Heuristics for Decreasing the Runtime

All the heuristics presented here are based on the types of edges. The following list shows the whole idea: (Let us call this list the *recipes for discard and contraction*.) In this list, for the first and third item, we change the initial graph via contraction along its certain parts, then apply our refinement method to the resulting reduced graph, and finally transform the graph back to what it has originally been. Having this done, we consider completing the proposed mappings by moving back to consideration the neglected parts during the period when the graph was in its contracted form. We will call this restoration of contracted vertices the *expansion phase*.

- *IS-A (rdfs:subClassOf)*: Contract all the paths into a pair of vertices between which there is an edge of type *IS-A*. The source of this edge will be the source of the original path, while the destination will be a new vertex, similarity of which is the maximum of the similarities of the original path excluding the source. At the expansion phase, consider this problem as an independent matching problem, but with the explanation after this list.

- *Disjoint (owl:disjointWith)*: If the difference between the distances of a concept in one ontology from a couple of disjoint concepts in another is above a certain threshold, remove the possibility of mapping the first concept to the one in the couple which is farther.

- *Equivalence (owl:equivalentClass)*: Contract all such vertices into one representing the whole group. Assign the maximum similarity of group to this new node. On expansion, there is no difference between different choices for matching between the two graphs.

- *owl:functionalProperty*: Functional properties should be mapped to functional properties, so,

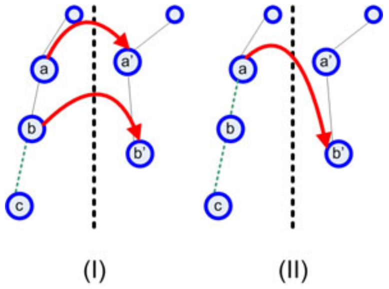

Fig. 8. Notes on expansion phase of IS-A.

discard all the alignments for which this does not hold.

- *rdfs:domain*: If there are two properties across the ontologies which domain over disjoint classes, discard all the alignments which map them to each other. Here, the "disjoint"-ness may be understood from several indicatives. For example, their distance may be more than a certain threshold. As an example of where an inference might also be involved, consider the question of mapping $p_1 \in O_1$ and $p_2 \in O_2$ to each other where p_1 and p_2 domain over C_1 and C_2 respectively, and where C_1 is *owl:disjointWith* C' while $d(C_1, C_2) > M$ (M being a certain threshold).
- *owl:intersectionOf*: Discard all the alignments that map classes which are intersections of disjoint classes. For instance, if $O_1 \ni C_1 = \cap_1^n C_{1i}$ and $O_2 \ni C_2 = \cap_1^m C_{2j}$, and we know that for some $i \in \{1, \ldots, n\}$ and some $j \in \{1, \ldots, m\}$, C_{1i} and C_{2j} are disjoint, we should be discarding all the alignments which map C_1 and C_2 to each other. ("disjoint"-ness, here, is meant to be what described for *rdfs:domain*.)

As far as the authors understand, all of the above heuristics should immediately seem rational except the first one. To have an intuition on the contraction, one can consider it like *Query Expansion* in the *Information Retrieval* [37] terminology. The expansion however is a little tricky. There is a fine observation which should be made on an *IS-A* paths:

Consider **Fig. 8(I)**, in which after expansion, it is chosen to map a to a', and b to b'. Here, there remains no choice for c. Now, consider **Fig. 8(II)**, in which a is mapped to b'. Note that because b *IS-A(n)* a, and b' *IS-A(n)* a', it is not correct to map b to a', and there remains no choice for either of b and c. With this scheme in mind, a solution to the expansion will become trivial, and the complexity of which will definitely be too small – say $O(n)$! However, we leave details of this until section 4.4 for a related discussion.

Table 2. Frequency of OWL (and RDF) properties.

Property	NoU	PI+	PI-
owl:incompatiblewith	0	0	0
owl:alldifferent	13	0.01	0.01
owl:differentfrom	13	0.01	0.01
rdfs:datatype	11	0	0.01
owl:symmetricproperty	27	0.01	0.02
owl:sameas	43	0.02	0.03
owl:equivalentproperty	70	0.03	0.05
owl:inversefunctionalproperty	100	0.04	0.08
owl:thing	233	0.09	0.18
owl:transitiveproperty	266	0.11	0.21
owl:oneof	313	0.12	0.24
owl:maxcardinality	807	0.32	0.63
owl:inverseof	932	0.37	0.73
owl:mincardinality	1315	0.52	1.02
owl:unionof	1629	0.65	1.27
owl:cardinality	2416	0.96	1.88
owl:allvaluesfrom	2841	1.12	2.21
rdfs:subpropertyof	2893	1.15	2.25
owl:equivalentclass	4836	1.91	3.76
owl:functionalproperty	7625	3.02	5.93
owl:disjointwith	7892	3.12	6.14
rdfs:domain	8476	3.36	6.59
owl:intersectionof	9482	3.75	7.38
owl:somevaluesfrom	22874	9.06	17.79
owl:restriction	53440	21.16	41.57
rdfs:subclassof	124005	49.1	—
Sum	252552	—	—
Sum without subclass of	128547	—	—

A question which may arise here is that "Why are there only a few properties chosen among the set of all *OWL* and *RDF* ones?" The reason behind this choice is a survey we have had on a set of 545 ontologies. **Table 2** shows the results of this survey (where *NoU* = Number of Usage, *PI+* = Percent of usage with *IS-A*, *PI-* = Percent of usage without *IS-A*). The authors believe that, according to that table, the percents of usage for the properties above *owl:equivalentClass* are not acceptable. However, we do not provide any heuristics for *owl:Resitriction* and *owl:someValuesFrom* either.

The reason why we do not offer any heuristics for *owl:Restriction* is that it is too general; The user may decide to use it for many reasons, yet there will be no guaranty that those reasons imply any degree of relevance for the properties they are restricting. It might be tempting to choose to discard the mappings which map restricted properties (the ones which are qualified with *owl:Restirction*) to non-restricted ones (the ones that are not qualified with *owl:Restirction*). This unfortunately will be wrong because whether or not the ontology decides to restrict a property can well be a mere matter of area of interest. For instance, an ontology describing plant transportation business may not be interested in the color of the plants they transport. On the other hand, a decoration company which is a customer of plant transportation seems to have such an interest. They obviously have common objects of interest (plants), but the latter chooses to restrict that object in their ontology while the former does not. And, discarding the mappings which map `Plant` to `Plant` across the ontologies is a mistake.

We also give no heuristics for *owl:someValuesFrom*. This policy takes place because we believe the W3C's

Algorithm 1. Naive approach.

```
1: **Input** $O$ and $O'$.
2: **Apply** a Threshold-Based Refinement on $O$ and $O'$.
3: **Apply** the recipes for Discard and Contraction on $O$, $O'$;
   call the resulting ontologies $O_1$ and $O'_1$, respectively.
4: Weight *all* remained possible mappings from $O_1$ to $O'_1$.
5: Expand back the contracted parts of $O_1$ and $O'_1$.
6: **Output** the mappings along with their weights.
```

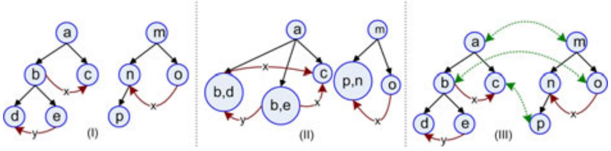

Fig. 9. The example, I- before, II- after contraction, III- final mapping.

description for this property restriction specifier [38] is rather tricky. Correctly understanding it will therefore need a fairly good understanding of Mathematical Logic, assumption of which for every ontology does not seem to be very realistic. The problem becomes more sever when one realizes that the mere phrase "some values from" does not inherently indicate any necessity for having the class it is describing to have the particular property that W3C describes.

4.2. Naive Approach

Algorithm 1 is a pseudo-code for mapping extraction based on the mapping scoring mechanism explained before. As an example of how this works, we consider **Fig. 9**.

Figure 9(I) shows the two ontologies O and O'. In accordance to this, **Table 3**-left shows d – the distance between concepts across them. Performing the third step of the Algorithm 1 will result in **Fig. 9(II)** and change of d accordingly as shown in **Table 3**-right. To clarify the values in the table we explain how the distance between (b,d) and (p,n) is calculated. According to **Table 3**-left, the distance between b and n is 0.9, similarly for b and p is 0.4, for d and n is 0.4 and finally for d and p is 0.6. Considering the Maximum of such values we reach to 0.9 for the distance between (b,d) and (p,n). Other values are calculated similarly.

Choosing $f(x) = x + 0.1$ and $g(x) = x$, **Table 4** will be the outcome of step 4. We explain how the values in row 1 are obtained. Since the edge between (b,e) and (b,d) is of type y for the edge between (o) and (p,n) is of type x (i.e. the edge is not preserved) therefore $w_0 = 0$. On the other hands $w_l = \overline{g}((b,d),(p,n)) + \overline{g}((b,e),(o))$ which is equal to $d((b,d),(p,n)) + d((b,e),(o))$ so we have $w_l = 0.9 + 0.5 = 1.4$. w_r is computed similarly.

Table 3. Distance of nodes before and after of contraction.

-	n	o	p
b	0.9	0.1	0.4
c	0.6	0.7	0.1
d	0.4	0.5	0.6
e	0.4	0.5	0.4

-	(p,n)	(o)
(b,d)	0.9	0.5
(b,e)	0.9	0.5
(c)	0.6	0.7

Table 4. The example after contraction.

	First pair	Second pair	w_0	w_l	w_r	Score
1	(b,d), (p,n)	(b,e) , (o)	0	1.4	1.4	−2.8
2	(b,e), (p,n)	(b,d), (o)	0	1.4	1.4	−2.8
3	(b,d), (p,n)	(c), (o)	0	1.6	1.6	−3.2
4	(c), (p,n)	(b,d), (o)	$\frac{1}{0.6}+\frac{1}{0.7}$	0	0	3
5	(b,e), (p,n)	(c), (o)	0	1.0	1.0	−2.0
6	(c), (p,n)	(b,e),(o)	$\frac{1}{0.6}+\frac{1}{0.7}$	0	0	3

In row 4 of the table we have a case where the edge is preserved. Therefore $w_0 = 1/(d((c),(p,n)) + 0.1) + 1/(d((b,d),(o)) + 0.1)$ which is equal to $1/0.6 + 1/0.7$.

As it is resulted from the table either of mappings 4 or 6 can be chosen as an ideal. This means that the problem is now reduced to two simpler subproblems: In the first, one should decide on mapping either of p and n to c, and, in the second, on choosing between b and d to be mapped to o. Considering the individual distances between vertices, one can easily choose to map b to o, and c to p. The extracted mapping, therefore, will be what is depicted in **Fig. 9(III)**.

4.3. GA-Based Mapping Extraction

One way to overcome the complexity of the naive approach is to treat the problem as one of optimization and then benefit from different approaches in that realm. Here, we briefly report the result of applying a GA approach to this problem, as detailed in [39].

As for similar GA solutions, we require a fitness function to evaluate each individual in our population, so we choose the coincidence-based weight function (Section. 3.3). We should define normalization functions and a distance metric to get a clear solution for fitness and individual evaluation. This distance function may either be a *string-based* distance or any other one. The distance between entities e_i and e_j, (i.e. $\delta(e_i, e_j)$), is considered to be the *Levenshtein distance* [40] of their labels. Normalization functions, f and g are then defined. $\overline{f}$ should be a positive decreasing function in for $d(v, m(v))$, so if $d(v, m(v))$ grows, it decreases to reduce the positive point. $\overline{g}$ should be a positive increasing function to grow with the growth of $d(v, m(v))$ to increase the negative point for that match. Normalization functions are defined by tuning the system.

$$f(v) = e^{\delta(v,m(v))}$$

$$g(v) = \frac{1}{e^{\max(5,15-\delta(v,m(v)))}}$$

Table 5. Head-to-head comparison EON 2004 tests between the competitors and GA.

Test	GA	Kar	UM	FUJI	Stan
201	0.40	0.43	0.44	0.98	1.00
202	0.38	n/a	0.38	0.95	1.00
204	0.74	0.62	0.55	0.95	0.99
205	0.48	0.47	0.49	0.79	0.95
206	0.67	0.48	0.46	0.85	1.00
221	(*)	n/a	0.61	0.98	0.99
222	0.74	n/a	0.55	0.99	0.98
223	0.79	0.59	0.59	0.95	0.95
224	1.00	0.97	0.97	0.99	0.99
225	0.98	n/a	0.59	0.99	0.99
228	(*)	n/a	0.38	0.91	1.00
230	0.85	0.60	0.46	0.97	0.99
301	0.85	0.85	0.49	0.89	0.93
302	0.83	1.00	0.23	0.39	0.94
303	0.68	0.85	0.31	0.51	0.85
304	0.85	0.91	0.44	0.85	0.97

These functions actually satisfy characteristics expected from $\overline{f}, \overline{g}$ explained above. $\overline{f}$ is a decreasing function and decreases with the growth of δ and g increases. Exponential functions are chosen for $\overline{f}, \overline{g}$ so that $\overline{f}, \overline{g}$ have close comparable values. These functions reflect discussions on positive and negative points for different categories of a coincidence-based weight.

In summary, fitness function $w(m)$ of section 3.3 is as follows:

$$\overline{f}(x) = e^{-\delta(x,m(x))}$$

$$\overline{g}(x) = e^{-\max(5,15-\delta(x,m(x)))}$$

$$\delta(x,m(x)) = \mathtt{LD}(label(x), label(m(x))$$

(LD = Levenshtein distance.) The next step is to design a crossover function to produce offspring – a new alignment – from two parents – two alignments. In the crossover function, single nodes are compared based on their weight. As described in [39], the weight of a single node in an alignment is the sum of weights of pairs in which that node is included. The best pairs among parents are chosen to be present in offspring.

Our first experiment with GA resulted in *precision* [37] of 0.7 when the two ontologies differ and 1 when they are the same. We conducted an experiment on a pair of Tourist ontologies [41] with population of 1,000 individuals, and the genetic algorithm converged in 32 iterations.

For our second experiment, we chose the EON 2004 [42] dataset, which contains tests for benchmarking the merit of OA algorithms [43]. We did not use the 1*xx* series because it was overly simple. **Table 5** shows the precision of this approach compared to that of the competitors of EON 2004 as reported in [44].

In **Tables 5** and **6**, Kar, UM, FUJI, and Stan stand for karlsruhe2, umontreal, fujitsu, and stanford teams. Cells marked with an asterisk indicate tests not applicable to coincidence-based approaches in general. Ref. [43] reports that for 221, “all subclass assertions to named classes are suppressed.” For 228, “properties and relations between objects have been completely suppressed.” GA outperforms karlsruhe2 and umontreal teams in 2*xx* tests while karlsruhe2 outperforms GA in the 3*xx* tests. In these – which according to [43] are the **real ontologies** – GA outperforms fujitsu.

Table 6. EON 2004 competitors vs GA.

Test	GA	Kar	UM	FUJI	Stan
2xx	0.70	0.59	0.54	0.94	0.99
3xx	0.82	0.90	0.37	0.66	0.92
total	0.73	0.71	0.48	0.89	0.97

Table 6 summarizes the comparison. Ref. [44] summarizes EON 2004 as follows:

> In this test, there are clear winners it seems that the results provided by Stanford and Fujitsu/Tokyo outperform those provided by Karlsruhe and Montréal/INRIA.
>
> In fact, it can be considered that these constitute two groups of programs. The Stanford+Fujitsu programs are very different but strongly based on the labels attached to entities. For that reason they performed especially well when labels were preserved (i.e., most of the time). The Karlsruhe+INRIA systems tend to rely on many different features and thus to balance the influence of individual features, so they tend to reduce the fact that labels were preserved.

Given that the concern of coincidence-based approaches, is generally not the mere labels attached to the entities, one can hardly say that they strongly rely on that. (One may argue that the types of the graphs are defined based upon the labels of the graphs. And, that is a correct observations. As explained throughout this paper, however, much more than labels, the coincidence-based approaches are mainly concerned with how typed graphs **coincide**.) We thus note that GA outperforms both Kar and UM.

GA approaches generally try to find near optimal solutions and not necessarily the global optimum. Because the run-time complexity of the naive approach limits its use to small ontologies, we have to rely on approximations such as those GA yields. Ref. [39] details this approach and explains how to keep the algorithm from falling into a local optima.

4.4. Approximative Approaches

Consider the idea of forming a new graph; a bipartite graph $G(V_1, V_2, E)$ where V_1 and V_2 are the sets of concepts of O_1 and O_2, respectively. An edge $e \in E$ is not typed but is weighted. This weight will show the mutual distance between its source and target (which is calculated in phase I). Applying a *Maximum Weight Matching* [27] for matching extraction here would be quite unwise because one would definitely lose the inherent structure and interrelationships of both ontologies. That is, regardless of the internal structure of the ontologies, the choice of an edge via this method would merely help to an overall optimization of the mutual distances between the concepts. In other words, there is no estimate of how much the resulted matching will also preserve the structure.

On the other hand, as described in Section 3.1, a method which merely considers structure is not precise enough either. As also explained in the same section, the notion of coincidence as a measure for knowing how coincident the two ontologies – **as a whole** – are becomes helpful. We explained further in Section 3.1 that the notion of coincidence goes hand-in-hand with alignments. Unfortunately, however, as discussed in Section 3.4, there is no knowledge at the moment about the complexity of the problem (of Typed Graph Isomorphism). We only know that it is as complex at least as the Graph Isomorphism problem itself.

The Naive Approach as offered in Section 4.2 is also already exponential. In Section 4.3, we tried to alleviate this complexity by exploiting GA which is no longer complex. However, knowing that excessive search is too complex and impractical, this section is about to offer a straight approach for coming up with a best-coinciding alignment. The question will then be: "Is there any approachable way for straightly finding an alignment which – although may not be the best coinciding – is **close to** that?" This is the question which the approaches we call *Approximative* will try to answer.

For finding the best-coinciding, the Naive Approach (in Section 4.2) tried to examine all the possible choices – which of course is an overkill. To find a close-to-best-coinciding, however, it will be nice if we can first have an **estimate** of how much pairing some arbitrary nodes may help in coming up with a better measurement for the degree of coincidence of the two ontologies. Once we have these pairwise estimates, we should next apply some minimization algorithms to minimize the overall distance too. This, as summerised in Algorithm 2, is in fact the sketch of **our** approximative approach.

In this section, only a quick discussion on how to apply the technique is presented. Especially, if the reader is interested to know why for the sake of minimizing the overall distance we do not use *Maximum Weigh Matching* [27], we suggest consideration of ibid. For short, we choose *Maximum Weight Non-crossing Matching* [45] over the former algorithm because the former may produce results which are conceptually wrong. As an example for **Fig. 8**, it may choose to map a to b' and in the same time c to a' – which, as discussed in section 4.1, will be wrong.

Algorithm 2. Sketch of the random walk approximation.

```
1: Input Ontologies $O_1(V_1,E_1)$ and $O_2(V_2,E_2)$ along
   with the metric space $(X,d)$ in which they are embedded.
2: Construct a bipartite graph $G(V_1,V_2,E)$ with
   weighted edges; each edge shows the helpfulness of
   pairing the endpoints of the edge for the two ontologies
   $O_1$ and $O_2$ to look more coinciding on $(X,d)$.
3: Apply Maximum Non-crossing Weight Matching to $G$.
4: Output the resulting heaviest matching.
```

Algorithm 3. Calculation of weights of E.

```
for all $v \in V_1 \cup V_2$ do {Initialisation}
    find the typical edge met along all $v$-stems.
end for
for all $v_1 \in V_1$ do {Evaluation}
    for all $v_2 \in V_2$ do
        Generate a random $v_1$-stem $s_1$; calculate $w_{s_1}(m(s_1))$
        Generate a random $v_2$-stem $s_2$; calculate $w_{s_2}(m(s_2))$
        $e(v_1,v_2) \leftarrow w_{s_1}(m(s_1)) + w_{s_2}(m(s_2))$
    end for
end for
```

It is natural to ask here: "How is the step 2 of Algorithm 2 done? And, where is the use of random walks?" Algorithm 3 is the way the weights for edges of E get calculated. We believe that this should answer both the above questions. There, considering s to be a randomly generated stem, $w_s(m(s)) = w_{s^+}(m(s)) - w_{s^-}(m(s))$, for which:

$$w_{s^+}(m(s)) = \sum_{\substack{u' \in m(s) \\ u' = m(u) \\ P_m(u,.)}} \left(\frac{1}{1-d(u,u')}\right)^{\mathfrak{d}^s(u',v') - \mathfrak{d}_{\overline{\mathrm{IS-A^S}}}(u',v')}$$

$$w_{s^-}(m(s)) = \sum_{\substack{u' \in m(s) \\ u' = m(u) \\ TP_m(u,.)}} d(u,u')^{-\mathfrak{d}^s(u,v)}$$

Here, for each v_1 and v_2, if s is a v_1-stem, $m(s)$ is the v_2-stem chosen to be best coinciding with s. And, finally, explanation of the symbols:

- In each iteration over the summation, u and u' are the current vertices. By $u' = m(u)$, we mean that u is from the pattern graph (O_1), and, is matched with u' from the target one (O_2).
- $d(.,.)$ is a function returning the distance between the concepts it is input with. Note that this distance is the metric of our metric space in fact.
- $\mathfrak{d}^s(.,.)$ is a function which takes two vertices of a graph, and returns the number of edges to be met along s for reaching to the second from the first. Note that this is independent of our metric space. In fact, this is applicable to any graph, yet, it is different from the common method of defining distance of vertices in graphs [46].

- $\mathfrak{d}_{\overline{IS-A}^s}(.,.)$ is the number of *IS-A* relationships met from the first argument toward the second when traversing on *s*.

- It is worth mentioning that $\mathfrak{d}(u',v') - \mathfrak{d}_{\overline{\mathrm{IS-A}}}(u',v')$ is always equal to $\mathfrak{d}(u,v)$. However, we prefer to retain the former because it better shows our purpose. Of course, for efficiency purposes, in practice, one may choose to use the latter over the former.

5. Conclusions and Future Works

We can summarize the novelty of this work as follows: First of all, the "coincidence" factor is something introduced for the first time in [47] by three authors of the current work. Secondly, up to our knowledge, the work reported in this paper is the first general formulation of the mapping extraction problem. Thirdly, other works either leave it completely to the user to extract the mappings, or do it in cooperation with the user, or do simple form of extraction.

The main contribution of this paper is to give a formal definition of the problem and our solution to respond to the problem. The paper also includes some experimental results on the GA implementation. The main merit of this approach is when it is applied for large ontologies where the structural relations play an important role in the alignment selection. For the case of simple ontologies where the decision making is mainly based on label similarities, other lexical based approaches might perform better.

We are now extending our research to find other approaches to use the coincidence-based weighting. One direction is to introduce Approximative Algorithms which perform a more elaborate random walk. Another direction is to consider other works in which Graph Theory and Metric Spaces are considered together, and find new ideas for further reduction of the size of the basic mechanism. One can consider [48] for example. Given that it is common to use Domain Theory [33] for evaluating the semantics of programming languages [49], we believe that there is a vast room for injecting those ideas in the realm of OM, especially in better adjustment of the partial order we were speaking about in this paper.

Acknowledgements

This research was in part supported by a grant from IPM (No. CS1385-4-01). Many thanks to Prof. Richard M. Wilson for his kind comments on typed graphs, Dr. Mohammad Mahdian for his notes on the heuristics on homeomorphism, and Taowei David Wang for his fertile data set containing the ontologies we have used here. Furthermore, we would like to give a thank to all the people at the Ontology and DL mailing list who helped. Last but not least, it remains to thank Prof. Alan Mycroft for reviewing our work.

References:

[1] J. Euzenat, "Towards composing and benchmarking ontology alignments," [Online]. Available: citeseer.ist.psu.edu/688410.html

[2] J. Euzenat et al., "State of the art on ontology alignment," Knowledge web NoE, deliverable 2.2.3, 2004.

[3] P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou, and S. Tessaris, "Specification of a common framework for characterizing alignment," Knowledge web NoE, deliverable 2.2.1, 2004.

[4] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Ontology matching: A machine learning approach," Handbook on Ontologies in Information Systems, Springer-Verlag, 2003.

[5] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches," Journal on Data Semantics, Vol.IV, 2005.

[6] "Call for papers – 4th european semantic web conference," 2007, [Online]. Available: http://www.eswc2007.org/callforpapers.cfm

[7] J. Hopcroft and R. Karp, "An n5/2 algorithm for maximum matchings in bipartite graphs," SIAM Journal on Computing, Vol.2, No.4, pp. 225-231, 1973.

[8] C. H. Papadimitriou and K. Steiglitz, "Combinatorial Optimization Algorithms and Complexity," Prentice-Hall, 1998.

[9] E. Rahm and P. Bernstein, "A survey of approaches to automatic schema matching," VLDB Journal, Vol.10, No.4, pp. 334-350, 2001.

[10] G. Bisson, "Learning in fol with similarity measure," in Proceedings of the 10th American Association for Artificial Intelligence conference, San-Jose (CA US), pp. 82-87, 1992.

[11] G. Schreiber, R. de Hoog, H. Akkermans, A. Anjewierden, N. Shadbolt, and W. V. de Velde, "Knowledge Engineering and Management," MIT Press, 2000.

[12] "Ontoweb. a survey on ontology tools. eu thematic network, ist-2000- 29243 deliverable 1.3, ontoweb — ontology-based information exchange for knowledge management and electronic commerce," available online: www.ontoweb.org/deliverable.htm, May 2002.

[13] P. Valtchev, "Construction automatique de taxonomies pour laide la reprsentation de connaissances par objets," Ph.D. Dissertation, Universite Grenoble, 1999.

[14] A. Maedche and V. Zacharias, "Clustering ontologybased metadata in the semantic web," in Proceedings of the 13th ECML and 6th PKDD, 2002.

[15] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), 1995.

[16] M. Ehrig and S. Staab, "Qom – quick ontology mapping," in Proc. ISWC-2003., 2003.

[17] M. Ehrig and Y. Sure, "Ontology mapping – an integrated approach," in 1st European Semantic Web Symposium (ESWS), pp. 76-91, 2004.

[18] J. Euzenat, J. Barrasa, P. Bouquet, and J. Bo, "State of the art on ontology alignment," Knowledge Web, Statistical Research Division, 2004.

[19] A. Doan, P. Domingos, and A. Halevy, "Learning to match the schemas of data sources: A multistrategy approach," Machine Learning, Vol.50, No.3, pp. 279-301, 2003.

[20] Y. S. M. Ehrig and S. Staab, "Bootstrapping ontology alignment methods with apfel," in Proceedings of the 4th International Semantic Web Conference (ISWC-2005), ser. Lecture Notes in Computer Science, pp. 186-200, 2005.

[21] H. Abolhassani, S. H. Haeri, and B. Hariri, "On ontology alignment experiments," Webology, Vol.3, No.3, 2006.

[22] R. Dieng and S. Hug, "Comparison of 'personal ontologies' represented through conceptual graphs," in 13th ECAI, Brighton (UK), pp. 341-345, 1998.

[23] S. Staab and A. Maedche, "Measuring similarity between ontologies," Lecture notes in artificial intelligence, No.2473, pp. 251-263, 2002.

[24] G. Stumme and A. Maedche, "Fca-merge: bottom-up merging of ontologies," in In Proc. 17th IJCAI, Seattle (WA US), pp. 225-230, 2001.

[25] A. V. Zhdanova and P. Shvaiko, "Community-driven ontology matching," in Proc. of ESWC, pp. 34-49, 2006.

[26] S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity flooding: a versatile graph matching algorithm," in Proc. 18th International Conference on Data Engineering (ICDE), San Jose (CA US), pp. 117-128, 2002.

[27] A. Gibbons, "Algorithmic Graph Theory," Cambridge University Press, 1985.

[28] P. Mitra, N. F. Noy, and A. R. Jaiswal, "Omen: A probabilistic ontology mapping tool," in 4th international semantic web conference (ISWC 2005), Vol.3729, pp. 537-547, 2003.

[29] H. L. Johnson, K. B. Cohen, W. A. Baumgartner, Z. Lu, M. Bada, T. Kester, H. Kim, and L. Hunter, "Evaluation of lexical methods for detecting relationships between concepts from multiple ontologies," Pac Symp Biocomput, pp. 28-39, 2006.

[30] J. Wang and L. Gasser, "Mutual online ontology alignment," in Proc. of the AAMAS 2002 Workshop, 2002.

[31] J. Li, "Lom: A lexicon-based ontology mapping tool," in Proceeding of the Performance Metrics for Intelligent Systems (PerMIS'04), Information Interpretation and Integration Conference (I3CON), Gaithersburg, MD., 2004.

[32] W. Rudin, "Principles of Mathematical Analysis," 3rd ed., New York: McGraw-Hill, 1976.

[33] S. Abramsky, "Domain Theory in Logical Form," 1987.

[34] R. Kothari, J. Basak, and I. Block, "Perceptually motivated measures for capturing proximity of web page elements: Towards automated evaluation of web page layouts," in The 11th International World Wide Web Conference, 2002.

[35] "Webster's New World College Dictionary," 4th ed., New York: Macmillan, 1998.

[36] W. A. Wilson, "On quasi-metric spaces," American Journal of Mathematics, Vol.43, pp. 675-684, 1931.

[37] R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval," Addison-Wesley, 1999, bAE r2 99:1 1.Ex.

[38] W3C, "Owl web ontology language guide, w3c recommendation 10 february 2004," Tech. Rep., 2004, [Online]. Available: http://www.w3.org/TR/2004/REC-owl-guide-20040210/#allValuesFrom

[39] V. Qazvinian, H. Abolhassani, and S. H. Haeri, "Coincidence based mapping extraction with genetic algorithms," in Proceedings of 3rd International Conference on Web Information Systems and Technologies (Webist 2007) Barcelona, Spain, March, 2007.

[40] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," Sov. Phys. Dokl., Vol.6, pp. 707-710, 1966.

[41] "Tourism ontology FOAM," available online: http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/ontologies/.

[42] "Call for papers – evaluation of ontology-based tools, 3rd international workshop," 2004, [Online]. Available: http://km.aifb.uni-karlsruhe.de/ws/eon2004/

[43] EON2004, "EON ontology alignment contest," 2004, [Online]. Available: http://oaei.ontologymatching.org/2004/Contest/

[44] "Evaluation of ontology-based tools, 3rd international workshop, table of results," 2004, [Online]. Available: http://oaei.ontologymatching.org/2004/Contest/results/

[45] F. Malucelli, T. Ottmann, and D. Pretolani, "Efficient labelling algorithm for the maximum non crossing matching problem," Discrete Applied Mathematics, Vol.47, pp. 175-179, 1993.

[46] D. West, "Introduction to Graph Theory (2nd ed.)," Upper Saddle River: Prentice Hall, 2001.

[47] S. H. Haeri, B. B. Hariri, and H. Abolhassani, "Coincidence-based refinement of ontology matching," in Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on advanced Intelligent Systems, 2006.

[48] B. Xiao, H. Yu, and E. Hancock, "Graph matching using spectral embedding and semidefinite programming," in Proceedings of the 15th British Machine Vision Conference, 2004.

[49] R. Tennent, "The denotational semantics of programming languages," Communications of the ACM, Vol.19, p. 437, 1976.

Name:
Seyed H. Haeri (Hossein)

Affiliation:
ULTRA Group, School of Mathematical and Computer Sciences, Heriot-Watt University

Address:
Office 1.68, ULTRA Group, Earl Mount-Batten Building, Riccarton, Heriot-Watt University, Edinburgh, UK

Brief Biographical History:
2006- Ph.D. Student in Theoretical Computer Science, Heriot-Watt University, UK

Main Works:
• S. H. Haeri, B. B. Hariri, and H. Abolhassani, "Coincidence-Based Refinement of Ontology Matching," Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on advanced Intelligent Systems (SCIS & ISIS 2006).

Membership in Academic Societies:
• Association of C/C++ Users

Name:
Hassan Abolhassani

Affiliation:
Web Intelligence Lab, Department of Computer Engineering, Sharif University of Technology

Address:
Azadi ave, Tehran, Iran

Brief Biographical History:
1989 Received Bsc. in Software Engineering from Esfahan University
1993 Received Msc. in Software Engineering from Sharif University of Technology
2001 Received Ph.D. in Knowledge Based Software Engineering from Saitama University
2003.9 Joined Computer Engineering Department of Sharif University of Technology
2003- Assistant Professor lecturing courses in Web Intelligence area and directing the Web Intelligence Laboratory

Main Works:
• ontology alignment, search engine result clustering, semantic search engines and trust in social networks

Membership in Academic Societies:
• Computer Society of IRAN

Name:
Vahed Qazvinian

Affiliation:
Web Intelligence Lab, Department of Computer Engineering, Sharif University of Technology

Address:
Azadi ave, Tehran, Iran

Brief Biographical History:
2003- B.Sc. student in Computer Engineering, Sharif University of Technology
2005- Senior Undergraduate Research Assistant at Semantic Web Lab under supervision of Dr. Abolhassani

Main Works:
• ontology alignment scoring and extraction

Name:
Babak Bagheri Hariri

Affiliation:
Web Intelligence Lab, Department of Computer Engineering, Sharif University of Technology

Address:
Azadi ave, Tehran, Iran

Brief Biographical History:
2007 Graduated in master of software engineering from Sharif University of Technology

Main Works:
• Semantic Web, ontology alignment