Webology, Volume 3, Number 3, September, 2006

On Ontology Alignment Experiments

Hassan Abolhassani
Assistant Professor, Sharif University of Technology, Tehran, Iran. E-mail: Abolhassani (at) sharif.edu

Babak Bagheri Hariri
Msc. Student, Sharif University of Technology, Tehran, Iran. E-mail: hariri (at) ce.sharif.edu

Seyed H. Haeri
Bsc. Student, Sharif University of Technology, Tehran, Iran. E-mail: shhaeri (at) math.sharif.edu

Received June 25, 2006; Accepted September 20, 2006

Abstract

Ontology Alignment is a process for finding related entities of different ontologies. This paper discusses the results of our research in this area. One of them is a formulation for a new structural measure which extends famous related works. In this measure with a special attention to the transitive properties, it is tried to increase recall with less harm on precision. Second contribution is a new method for compound measure creation without any need to the mapping extraction phase. Effectiveness of these ideas is discussed and quantitative evaluations are explained in this paper.

Keywords

Ontology alignment; Structural measure; Compound measure; Lexical measure; Sensitivity analysis

---

1. Introduction

Like the Web, the *semantic web* is distributed and heterogeneous. To support interoperability and common understanding between the different parties, ontologies are used. Since there is no expectation to have a limited number of ontologies, *Ontology Alignment* is needed. It is used for finding semantic relationships among the entities of ontologies. Many of the existing methods for ontology alignment compare similarity of entities using some predefined measures, and via the interpretation of the results, they put forward some possible set of semantic relationships among the entities.

The measures for similarity computation can be divided into two general groups; namely, "*Lexical Measures*" and "*Structural Measures*." Lexical measures are based on surface similarities such as that of the title, label, or URI of entities. The main idea in using such measures is the fact that it happens that usually similar entities have similar names and descriptions across different ontologies. On the other hand, structural measures try to

recognize similarities by considering the kinship of the components and structures residing in the ontology graphs. Leveraging other available information in two ontologies, they hope to recognize related entities outside the site of the lexical measures. Methods which are used by such similarities rely on the intuition that elements of two distinct models are similar when their adjacent elements are similar (Euzenat, 2004). Existing work uses the following seven criteria for deciding that two entities are similar:

- C1: Their direct super-entities (or all of their super-entities) are already similar (Dieng, 1998).
- C2: Their sibling-entities (or all of their sibling-entities) are already similar.
- C3: Their direct sub-entities (or all of their sub-entities) are already similar (Dieng, 1998).
- C4: All (or most) of their descendant-entities (entities in the subtree rooted at the entity in question) are already similar.
- C5: All (or most) of their leaf-entities (entities, which have no sub-entity, in the subtree rooted at the entity in question) are already similar (Madhavan, 2001).
- C6: All (or most) of entities in the paths from the root to the entities in question are already similar (Bach, 2004).
- C7: All (or most) of relative entities to the entities in question using properties are similar.

In this paper a new measure for structural similarity of two entities (i.e. concepts) from two given ontologies is presented. Another developed idea is to combine measures and create a new compound measure with the hope that there would be possible to create better mappings. In this paper, we report our new idea for this task which is based on an artificial neural network model.

In section 2 a review of related works is given. Section 3 introduces our proposed structural measure together with evaluations on it. Section 4 discusses about our new idea for compound measures' creation. Finally a conclusion is given in section 5.

2. Related works

In section 2.1., a survey of works related to structural measures are given and is followed by a survey on compound measures creation in section 2.2.

2.1. Works on structural measure

There have been numerous works for finding structural similarities of graph entities. Some of them are developed specifically for ontology alignment while some others have been developed for other domains, like for WordNet (wordnet.princeton.edu) similarity, but still are useful for the ontology alignment problem.

2.1.1. Structural Topological Dissimilarity on Hierarchies

This method (Valtchev, 1997) computes the dissimilarity of elements in a hierarchy based on their distance from closest common parent. Structural topological dissimilarity δ: $O \times O \rightarrow \mathbf{R}$ is a dissimilarity over a hierarchy $H = \langle O, \leq \rangle$, such that:

$$\forall e,e' \in O,\ \delta(e,e') = \min_{c \in O} [\delta(e,c) + \delta(e',c)] \qquad (1)$$

Where δ(e,c) is the number of intermediate edges between an element and another element c. This corresponds to the unit tree distance of (Barthlemy & Gunoche, 1992) with weight 1 on each edge.

2.1.2 Upward Cotopic Similarity

The Upward Cotopic distance (Maedche, 2002) δ: O ×O → **R** is a dissimilarity over a hierarchy H ‹ O, ≤›, such that:

$$\delta(c,c') = \frac{UC(c,H) \cap UC(c',H)}{UC(c,H) \cup UC(c',H)} \quad (2)$$

UC(c,H)={c′ ∈ H; c ≤ c′} is the set of superclasses of c.

2.1.3 Similarity Distance

This measure (Zhong, 2002) computes the relationship among entities for a single hierarchy. The concept similarity is defined as:

$$Sim(c_1,c_2) = 1 - distance(c_1,c_2). \quad (3)$$

Every concept in the concept hierarchy is assigned a *milestone* value. Since the distance between two given concepts in a hierarchy represents the path over the closest common parent ccp, the distance is calculated as:

$$distance(c_1, c_2) = distance(c_1, ccp) + distance(c_2,ccp) \quad (4)$$

$$distance(c, ccp) = milestone(ccp) - milestone(c) \quad (5)$$

The milestone values of concepts in the concept hierarchy are calculated as follows:

$$milestone(n) = \frac{1}{k^{l(n)+1}} \quad (6)$$

where l(n) is the length of the longest path from the root to the node n in the hierarchy and k is a predefined factor larger than 1 indicating the rate at which the milestone values decrease along the hierarchy.

2.1.4 Resnik Similarity

This method (Zhong, 1995) introduces a measure to calculate similarity of WordNet 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:

$$sim(c_1,c_2) = \max [-\log p(c)] \quad (7)$$

$$c \in S(c_1,c_2)$$

$$p(c)=\frac{freq(c)}{N} \quad (8)$$

In the above formula $S(c_1,c_2)$ is the set of concepts that subsume both c_1 and c_2, freq is the number of occurrences of a concept in a hierarchy and N is the total number of concepts. When it is applied in a single ontology, freq should be interpreted as the number of children for the concept.

Similarly methods like those introduced in (Kalfoglou, 2005), (Doan, 2003), and (Ehrig, 2004) also try to use the similarity of parents, children and siblings to calculate the relationships of concepts in two ontologies.

All the above-mentioned measures cannot be applied as such in the context of ontology alignment since the ontologies are not supposed to necessarily share the same taxonomy. For that purpose, it is necessary to extend these kinds of measures over a pair of ontologies. For example in (Valtchev, 1999) and (Euzenat, 2004), this amounts to use a (local) matching between the elements to be compared.

2.1.5 Anchor Prompt

This measure (Noy, 2001) tries to find relationships between entities based on the primary relationships recognized before. The central observation behind Anchor-Prompt is that if two pairs of terms from the source ontologies are similar and there are paths connecting the terms, then the elements in those paths are often similar as well.

2.1.6 Similarity Flooding

The Similarity Flooding (SF) (Melnick, 2002) compares graphs representing the schemas, looking for similarities in the graph structure. SF utilizes a hybrid matching algorithm based on the ideas of similarity propagation. The basic concept behind the algorithm is the similarity spreading from similar nodes to the adjacent neighbors through propagation coefficients.

2.1.7 OLA

OLA (OWL Lite Aligner) (Euzenat, 2003) is designed with the idea of balancing the contribution of each component that compose an ontology. OLA converts definitions of distances based on all the input structures into a set of equations. The algorithm then looks for the matching between the ontologies that minimizes the overall distance between them.

Table 1: Comparing Different Structural Similarity Measures

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	OA
ST	√		√					
UC	√							
SD	√		√					
RS	√							
AP	√		√		√	√		√
OL	√	√	√	√	√	√	√	√

SF	√	√	√	√	√	√	√	√
IC	√		√	√	√	√	√	√

Table 1 shows a comparison of the methods based on the types of information they use. *ST* is an abbreviation for *Structural Topological*, *UC* is for *Upward Cotopic*, *SD* is for *Similarity Distance*, *RS* is for *Resnik Similarity*, *AP* is for *Anchor Prompt*, *OL* for *OWL Lite Aligner* and *IC* is for our proposed method discussed later named as *Information Content*. Also C_1 to C_7 are described in Section 1, and *OA* is an abbreviation for *Ontology Alignment Specific* which shows if the method is designed specially for ontology alignment.

2.2 Compound Measure Creation

In this section, we briefly review famous works for compound measure creation.

Let O be a set of objects which can be analyzed in n dimensions. Here each dimension represents a measure. Then the Minkowski distance (Euzenat, 2004) between two such objects is:

$$\forall x,x' \in O,\ \delta(x,x') = \frac{\sum_{i=1}^{n} \text{delta}(x,x')^p}{p} \tag{9}$$

In which $\delta(x_i,x'_i)$ is the dissimilarity of the pair of objects along the i_{th} dimension. Therefore having a set of distance measures we can combine them this way to a compound distance measure.

Another approach is to use the weighted sum (Euzenat, 2004) between two such objects:

$$\forall x,x' \in O,\ \delta(x,x') = \sum_{i=1}^{n} w_i \times \delta(x_i,x'_i) \tag{10}$$

Also we can consider the weighted product as below:

$$\forall x,x' \in O,\ \delta(x,x') = \prod_{i=1}^{n} \delta(x_i,x'_i)^{\lambda_i} \tag{11}$$

There are also learning-based methods. In this group of methods, using machine learning techniques, some coefficients for weighted combination of measures are attained. Optimal weights in such methods are calculated by defining or proposing some specific measures and applying them on a series of test sets - an ontology couple with actual mappings between their elements.

One of such methods is Glue (Doan, 2003). Glue use machine learning techniques to find mappings. It first applies statistical analysis to the available data. Then it generates a

similarity matrix, based on the probability distributions, for the data considered and uses "constraint relaxation" in order to obtain an alignment from the similarity. Euzenat et al. (2004) use a set of basic similarity measures and classifiers each operating on different schema element characteristics. These classifiers provide local scores which are linearly combined to give a global score for each possible tag. The final decision corresponds to the mediated tag with the highest score. Combining the different scores is a key idea in their approach.

The work closest to ours is probably that of Ehrig et al. (2005). In *APFEL* weights for each feature is 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.

3 Our Semantic Measure

We first introduce a new measure, and then in Section 3.2, we present its theoretical and intuitive basis, and finally discuss evaluation results in Section 3.3.

3.1 Definition

The purpose of this measure is to have a means to calculate structural similarity between the entities of two given ontologies. In this measure, similarity among entities of two ontologies is estimated using a real number based on existing transitive relationships across the ontologies.

Our measure is, in fact, deemed to be a generalization for the Resnik Hierarchical Similarity (Zhong, 1995). As explained, the presented method in Resnik is not directly usable for the ontology alignment problem. Therefore, here we try to customize it so that it can be applicable on ontologies. The first customization is generalizing the concept of **Common Father** to a concept applicable for a pair of ontologies. We do this by identification of similar entities across the two ontologies. In order to propose a structural similarity, we need to somehow identify some similar pairs of entity. We perform this alike the other methods - such as Anchor-Prompt (Noy, 2001) - in which for semi-automatic approaches the pairs are inputted from the user, and for the automatic ones the lexical similarities are employed. Having pairs with similarity above a certain threshold, we are ready to identify the related pairs of concept.

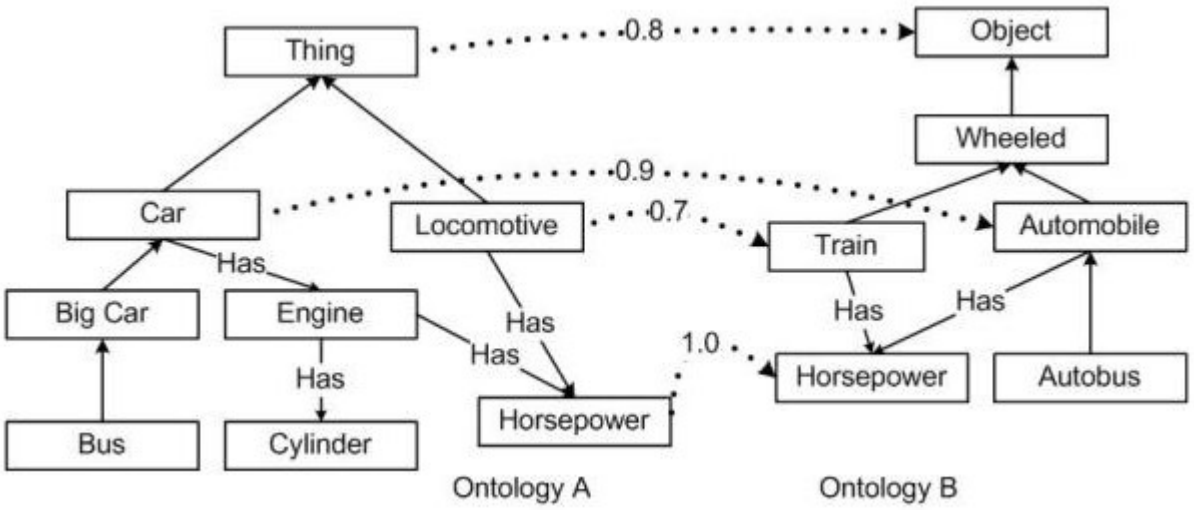
One of the conceptual heterogeneity types is the granularity (Bouquet, 2005). Granularity Heterogeneity occurs when one ontology has more details than the other. In such situations, there might exist some entities in one ontology which are out of consideration in the other. Existence of such entities in one side, may make the identification of structural relationships across entities of ontologies quite problematic. One of the goals of presenting this measure is enabling current ones in resisting against such effects. For this purpose, a concept called *Relative Elements* is introduced which is considered to be a generalization of the closest common father. Relative elements of a pair (c_1,c_2) from ontologies O_1 and O_2 are defined as a pair (rc_1,rc_2), such that following requirements are satisfied:

1. $rc_1 \in entities(O_1)$ and $rc_2 \in entities(O_2)$.
2. rc_1 and rc_2 are already identified to be similar either from the user inputs or from one of the lexical similarity measures, for which the amount of similarity is greater than a certain specified threshold.
3. If we represent ontology entities as nodes of a graph, and its properties as directed edges from its domain to its range, and also represent subclass relationship (i.e. *is-a* relationship) as an edge from the special to the general side, there exists at least one

path from c_1 to rc_1 or from rc_1 to c_1. The same property holds likewise for c_2 and rc_2. A vector first element of which is the direction of a path (indicated using 0/1 here), and the other elements of which are the properties met along the path is what we call *Relationship*. If the direction is from the entity to its corresponding relative entity, we represent it by 0, otherwise by 1. For instance, in ontology A of Figure 1, Bus has the ⟨1·is-a·is-a·has·has⟩ with Horsepower.

4. *Relationships* between c_1 and rc_1 holding also for c_2 and rc_2 are of length greater than 1 and less than the predetermined value MaxLength. Also, in such a path there is no repeated entity - there is no cycle in it.

Figure 1: Example of Ontology Alignment



5. There exists at least one *Optimal Relationship Couple* for c_1 and c_2 connected via rc_1 and rc_2. *Optimal Relationship Couple* consists of two *relationships*, one from c_1 to rc_1, and another from c_2 to rc_2 such that:
    a. Their *Reduced Relationships* are equal.
    b. Their total lengths among the pair relationships satisfying the first condition, is minimum.

We define the sum of the lengths of these two vectors as length between c_1 and c_2, connected via rc_1 and rc_2.

Reduced Relationship of a given *relationship* is a vector whose the first element (i.e. direction indicator) is as same as the relationship, and its other elements are the same of the relationship except that all the *is-a* properties are eliminated, and every run of transitive equal properties is replaced with only one occurrence of such a property. As an example, in Figure 1 several *relationships* and their *reduced relationships* are shown.

Figure 2: Extracting *Reduced Relationship* from *Relationship*

R.	$0 \cdot T_1 \cdot T_2 \cdot T_2 \cdot N_1 \cdot N_1 \cdot N_2 \cdot T_1 \cdot I \cdot T_3$
R. R.	$0 \cdot T_1 \cdot T_2 \cdot N_1 \cdot N_1 \cdot N_2 \cdot T_1 \cdot T_3$
R.	$0 \cdot T_1 \cdot T_2 \cdot N_1 \cdot N_1 \cdot N_2 \cdot T_1 \cdot T_3 \cdot T_3$
R. R.	$0 \cdot T_1 \cdot T_2 \cdot N_1 \cdot N_1 \cdot N_2 \cdot T_1 \cdot T_3$
R.	$1 \cdot T_1 \cdot N_1 \cdot I \cdot I \cdot N_2 \cdot T_1 \cdot T_2 \cdot T_2 \cdot I$
R. R.	$1 \cdot T_1 \cdot N_1 \cdot N_2 \cdot T_1 \cdot T_2$

T_x:	Transitive Property	*R.*:	Relationship
N_x:	Non-Transitive Property	*R. R.*:	Reduced Relationship
I:	Is-a		

Now, we define structural similarity δ: $O \times O \rightarrow \mathbf{R}$ between two entities c_1 and c_2 as follows:

$$\delta(c_1,c_2) = \sum_{(r_1,r_2) \in RE(c_1,c_2)} \frac{sim(rc_1,rc_2)^{\alpha} \times IC(rc_1,rc_2)}{length^{\beta}} \quad (12)$$

$$IC(c_1,c_2) = \sqrt{logp(c_1) \times logp(c_2)} \quad (13)$$

In which c_1 and c_2 are two entities from two ontologies in consideration. α and β are real numbers and have to be tuned. sim:$O \times O \rightarrow \mathbf{R}$ is the lexical similarity for two entities (each from a different ontology) which can be determined by one of the existing measures (e.g. string-based similarities, WordNet similarities or their combination). Function IC: $O \times O \rightarrow \mathbf{R}$ represents the information content for relative entities in which $P:O \rightarrow \mathbf{R}$ is a function that returns a number between 0 and 1 for a given entity based on its location in the ontology. Here, we extend the concept of common father from Resnik to a pair of similar concepts as shown in formula 12. For calculating function P, first we define function freq. This function gets one as input and returns that entity's number of children as output. Now, we define function P as follows:

$$p(c) = \frac{freq(c)}{N} \quad (14)$$

In which N is total number of entities in the ontology. Also, value of length for c_1 and c_2 regarding rc_1 and rc_2 is computed from item number 5 of relative entity definition.

3.2 Theoretical and Intuitive Basis of the Proposed Measure

As mentioned earlier, various measures for identifying relationships in structures related to ontologies have been introduced. Some of them are designed specifically for ontology alignment. For instance, Anchor-Prompt by recognition of some anchor points and finding paths between anchor points with similar length in the two ontologies tries to assign higher weights to the elements of such a path so that it be possible to identify some relationships that are not recognizable by lexical measures. Because of the limitation of the equal lengths, this method is only able to identify new semantic relationships among those entities that have really the same structures and this is not something happening most of the time. For instance, if we have $a \rightarrow b \rightarrow c \rightarrow d$ in one ontology and $a \rightarrow x \rightarrow d$ in another one, this algorithm does little effort in identifying relationships between x and b or c so to not decrease the precision. In other words, it will sacrifice recall for precision.

One of our goals in presenting the new measure is to study this kind of relationships in more details so we will be able to increase recall as much as possible with minimum decrease of precision. In this measure with a special attention to the transitive properties and more specifically to *is-a* relationships (i.e. subclass-superclass), we try to increase recall with less harm on precision. As an example in $a \rightarrow b \rightarrow c \rightarrow d$, if we assume each of the arrows as an *is-a* relationship and if in the second ontology we have $a \rightarrow x$, so that an entity of two ontologies are matched, then matching probability for each of entities b, c or d from first ontology and x in second ontology is the same and there is no difference

between them in terms of matching probability. Also this characteristic holds for other transitive properties. Suppose arrows be consist – of relationship, in this case if a has a relationship with b and b has a relationship with c, we can infer that a has relationship with c too. Now if in the second ontology, a has a relationship with x, the probability for each of entities b, c or d being matched to x is equal. This is in fact the basis for the concept of *reduced relationship* in our measure. In this vector, we eliminate all *is-a* properties because for the mentioned reason, their existence and difference between two vectors in them and number and location of these properties, none of them decrease the probability of matching two entities of ontologies.

For the same reason, we replace other consecutive transitive equal properties by one of them, because their repetition cannot make any difference in entities matching probability. However, we should keep non-transitive properties in *reduced relationship* as they appear in the original vector. As a result, this method similar to Anchor-Prompt checks that vectors are at the same length and even with the same signature, but after construction of *reduced relationships* there are more chances to have structural matching than can be found by Anchor-Prompt.

In (Zhong, 1995), according to quantifying information content in the nodes, it is tried to identify structural similarities. One of the consequences of this idea is the fact that the degree of structural similarity between two entities is not affected by the amount of entities' distance from their *Closest Common Father*. Justification of this idea is the transitive property of *is-a* mentioned above. On the other hand, in (Zhong, 2002) the distance is considered to be important. Regarding our generalization of common father, a new definition for length is induced. For simplifying this concept in a case that other relationships besides *is-a* are considered in ontology, this concept becomes totally matching with length concept in (Zhong, 2002). In the presented formula for finding the degree of similarity, we affected the length by power of β. In a case that this value is small and close to zero, behavior of this measure will approach the behavior of *Resnik Similarity* (Zhong, 1995), otherwise to the behavior of *Similarity Distance* (Zhong, 2002).

In the formula 12, IC(rc1,rc2) exists which shows the fact that the more number of children of one entity, the less matching probability of the children. This definition is in fact a generalization of the information concept defined in (Sure, 2004) and (Ross, 1976). Here, for simplicity, we took into the consideration the classic definition of child concept as entities which has *is-a* relationships with a specified entity under consideration. However, we can also generalize this concept to the entities for which there is a path from the specified entity to them.

Expression $sim(c_1,c_2)^{\alpha}$ in the formula 12 is also a number always between threshold $^{\alpha}$ and 1. If we set the value of α to zero, then lexical similarity will not directly affect the value of structure similarity and it only gets checked to be above a specific limit. If we set the value to 1, lexical similarity will have the direct effect on the amount of structure similarity. This way, by adjusting the value of α, we can get a desired behavior from structure similarity.

In the presented measure, if we have two entities of a single ontology as input, meaning $O_1=O_2$, and we set the β to 0 and use max instead of ∑, then our measure will be similar to the measure introduced in (Zhong, 1995) for comparing the entities in WordNet. Also, if the value of *IC* is ignored and $O_1=O_2$ and only *is-a* relationships are taken into consideration, our measure is similar to the measure introduced in (Zhong, 2002).

With this measure, 6 criteria out of 7 criteria mentioned in section 1 are covered and only sibling's entities are not covered just because the limitations that we put in the conditions 2 and 3 of the *relative entities* definition - paths should be traversed in one direction. Other

information such as indirect children, fathers and leaves all can have influence in identifying degree of similarity of two entities.

As an example in Figure 1, we calculate the degree of similarity between Bus and Autobus by these assumptions: $\alpha = 1$, $\beta = 0.5$, threshold = 0.7 and also the assumption that all the lexical similarities which are not shown in the figure have the value less than threshold=0.7. First, we calculate relative elements (c_1,c_2). We know from the first condition of relative elements definition, that rc_1 is one of the entities of 'A' ontology, and rc_2 is one of the entities of 'B' ontology. From the second condition of the definition, it is deduced that rc_1 is a member of {Thing, Horsepower, Car, Locomotive} and rc2 is a member of {Object, Automobile, Horsepower, Train}. From the condition 3, it is inferred that rc_1 cannot be Locomotive because there is no path from c_1 to Locomotive or from Locomotive to c_1, also same thing holds for rc_2 and Train. In this stage, *relationships* shown in Table 2 are identified.

Table 2: Calculating *Reduced Relationship* Vector

Ontology A			
c_1	rc_1	Relationship	Reduced Relation.
Bus	Thing	1 · is-a · is-a · is-a	1
Bus	Car	1 · is-a · is-a	1
Bus	Horsepower	1 · is-a · is-a · has · has	1 · has
Ontology B			
c_2	rc_2	Relationship	Reduced Relation.
Autobus	Object	1 · is-a · is-a · is-a	1
Autobus	Automobile	1 · is-a	1
Autobus	Horsepower	1 · is-a · has	1 · has

Table 3: Calculating Optimal Relationship Couple

rc_1,rc_2	$Relationship_1$	$Relationship_2$	Length
Hors., Hors.	1 · is-a · is-a · is-a	1 · is-a · is-a · is-a	6
Car, Auto.	1 · is-a · is-a	1 · is-a	3
Thing, Object	1 · is-a · is-a · has · has	1 · is-a · has	6

All these six vectors satisfy condition 4. Regarding condition 5, corresponding *reduced relationships* for the above six vectors are also shown in Table 2. By considering Thing and Object as relative entities, ‹1·is-a·is-a·has·has, 1·is-a·has› is one *optimal relationship couple* between Bus and Autobus with a length equal to 4+2=6. Table 3 shows *optimal relationship couples* between Bus and Autobus which are gained by considering ‹Thing, Object›, ‹Car, Automobile›, ‹Horsepower, Horsepower› as relative entities. The values of *Information Content* are calculated as follows:

$$IC(Thing,Object) = \sqrt{\log 8/8 \times \log 6/6} = 0.000$$

$$IC(Car,Automobile) = \sqrt{\log 3/8 \times \log 2/6} = 0.451$$

$$IC(Horse.,Horse.)= \sqrt{\log 1/8 \times \log 1/6} = 0.838$$

Now we can calculate the amount of proposed similarity:

$$\delta(Bus,Autobus)= \frac{0.9^{0.5} \times 0.45}{3^{0.5}} + \frac{0.838}{6^{0.5}} = 1.12$$

It should be noted that this value and other calculated values must be normalized before interpretation and usage.

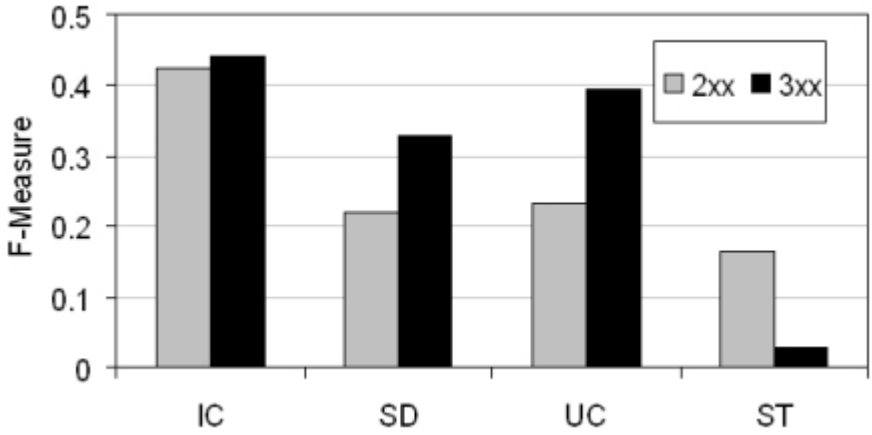
3.3 Evaluation of Measures Using Precision and Recall

To evaluate the performance of the proposed measure, we used EON_{2004} (Sure, 2004) data set - tests numbers 203, 221, 222, 223, 230, 303, and 304 - and compare our measure with *Upward cotopic*, *Similarity Distance* and *Structural Topological* measures. As discussed in section 3.1 our measure uses several parameters for adjustments. We estimate appropriate values for the parameters in our measure using a new evaluation method which is based on *Sensitivity Analysis* of Data Mining area. Also we evaluate our measure using Precision and Recall as well as the new evaluation method.

We have developed a simple framework using Jena[3] which by having two ontologies as input, it compares elements of the first ontology with all elements of the second one based on all mentioned measures. To do so, Lexical Similarities are computed based on average of normalized *Levenshtein* (Levenshtein, 1966) and *Resnik* (Zhong, 1995) similarity values. Two concepts are considered to be *Lexically Similar* if they satisfy following: (1) Lexical Similarity value for them is greater than a specified threshold - here we set it to be 0.5. (2) Each concept is matched with at most one concept in the other ontology. Such relationships are basic information for calculating structural similarity.

In each experiment, one of the structural measures are selected and *Precision* and *Recall* values as well as their harmonically average - *F-Measure* - are calculated. Figure 3 shows the results in which IC is an abbreviation for *Information Content* achieved by setting alpha=beta=0 and MaximumLength=12 (next section discusses about why such values are selected). UC is an abbreviation for *Upward Coptic Distance*, ST is for *Structural Topological* and SD is for *Similarity Distance*. Also 2xx indicates the average results for 205, 221, 222, 223 and 230 tests. Similarly 3xx is average results for 302, 303 and 304 tests.

Figure 3: Evaluation of Structural Measures Using Precision and Recall



As shown in the figure, *Information Content* measure has shown better behavior in both two test sets. It should be noted that the figure shows *F-Measure* for only structural similarities. If we consider lexical similarities as well, then the *F-Measure* values are much higher than what is shown. However, since we are interested on comparing structural similarity measures we do not include lexical similarity in the calculation of *F-Measure* here.

4. Compound Measure Creation by a Neural Networks based model

It is customary to have an evaluation on measures to calculate their weights in a compound measure. Such evaluations are normally based on Precision and Recall computations. However, to use Precision and Recall it is necessary to perform mapping extraction. Such a task depends on the definition of a **Threshold** value, as well as the approach for extracting, and some other pre-defined constraints. Such dependencies results in in-appropriateness of current evaluation methods.

We propose a new method for evaluation of measures and creating a compound measure from some of them without any need to the mapping extraction phase. Like other learning-based methods, it needs an initial training phase, in which an ontology pair with actual mappings in them is fed in to the algorithm. A few measures, along with their associated *category* are also considered. A category represents measures which share similar processing behaviors. For example, each of *String Measures*, *Linguistic Measures*, *Structural Measures* and so on are considered as a category. Our proposed algorithm selects one measure from each category. Therefore, if it is intended to be used on a specific measure, we can define a new category and introduce the measure as its mere member so far.

Our aim behind defining categories and assigning measures to them is that, in combining measures, usually String and Linguistic based measures are more influential than others, and, therefore if we do not use such a categorization, and apply the algorithm on a set of uncategorized measures, most of the selected ones are linguistic-based, and which results in a lower performance and flexibility of algorithm on different inputs.

Having measures and their associated categories, the algorithm selects the best measure from each category and proposes an appropriate method to aggregate them. To do this a data mining approach is considered. Therefore, we need to formulate the problem in a way that a Data Mining algorithm can be applied on. For this purpose, we operate as mentioned in the following sections.

4.1 The method

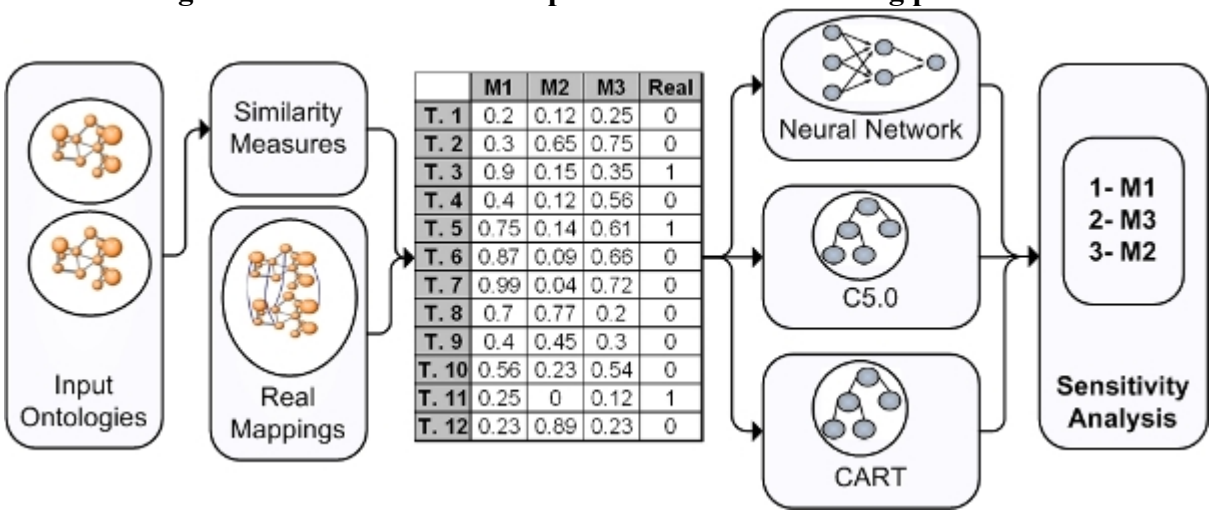
One of the customary problems in Data Mining area is to create a model for calculating values of a variable named as *Target Variable* based on the values of some other variables referred to as *Predictors*. In supervised-based learning methods, having a suitable training set, the model is constructed. Various approaches have been developed in this regard. The one which is used in this paper for the Ontology Alignment problem is based on neural networks. The idea stems from the fact that in Ontology Alignment we have a number of measures acting as predictors, and the goal is to find their importance or effects on the target variable - which turns out to be the actual mappings across ontologies. Such an interpretation reduces the alignment measure evaluation problem to a data mining one. The detail of the approach is as follows:

For a pair of ontologies, a table is created with cells showing values of a certain (set of) comparison measure(s), of an entity from the first ontology to an entity from the second. For each pair of elements across the ontologies, and for each measure for finding mappings, we associate a number which is the predicate of that measure on the similarity (or distance) between the pair. We present this in a table rows of which stand for the pairs, while its columns stand for the measures. There is a further column in this table which shows whether or not there exists a mapping between the pair *in the real world*. The cells of this final column will be either 0 or 1, based on the existence of such a mapping.

All of such tables are aggregated in a single table. In this final table the column representing actual mapping value between a pair of entities is considered as the target variable and the rest of columns are predictors. The problem now is a typical data mining and then we can apply classic data mining techniques to solve it. Figure 4 shows the process.

In this figure, the proposed method is shown. In it, *Similarity Measures* represents measures being used. Also *Real Mappings* are actual mappings between entities of input ontologies which are obtained from train set. The middle table is constructed as explained before in which m_1, m_2 and m_3 are values from different measures and the last column, real, is the actual mapping between two entities. *Neural Networks*, $C_{5.0}$ and *CART* are models which are used to find the most influential measures. Right oval shows the results obtained from different models with numbers showing the priority value of each measure.

Figure 4: Formulation of the problem as a Data Mining problem



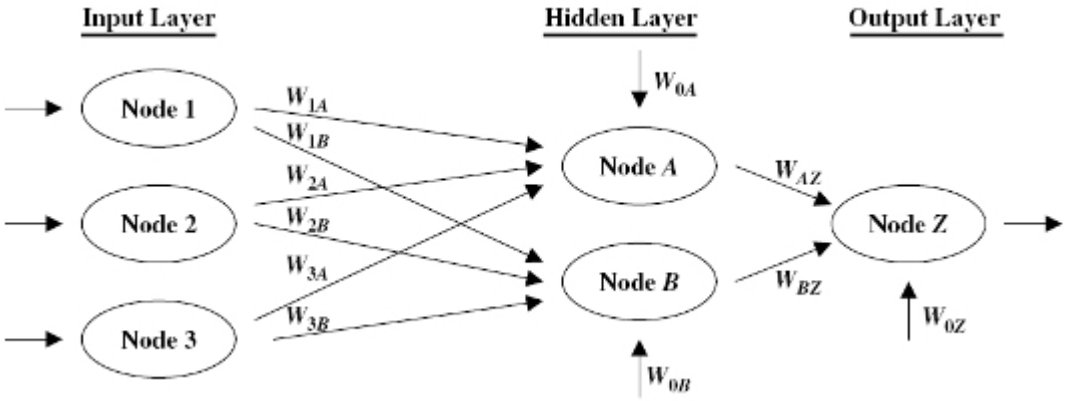
As suggested in the figure we can apply any learning-based model like *neural networks* (Larose, 2005), $C_{5.0}$ and CART (Breiman, 1984) decision trees. However in our

experiments neural network model has shown the better response and therefore we explain its results in this paper.

Figure 5 shows a sample neural network model for this problem. Inputs to the network are values of measures (for example M1, M2 and M3 in the Figure 4). The output of the network having the real values in each row of the table *neural network* training is done to find appropriate weights.

A *neural network* consists of a layered, feed forward, completely connected network of artificial neurons, or nodes. The neural network is composed of two or more layers, although most networks consist of three layers: an input layer, a hidden layer, and an output layer. There may be more than one hidden layer, although most networks contain only one, which is sufficient for most purposes. Figure 5 shows a *neural network* with three layers. Each connection between nodes has a weight (e.g., W_{1A}) associated with it. At initialization, the weights are randomly assigned to values between 0 and 1.

Figure 5: The Neural Network Model



After the training is complete, a *Sensitivity Analysis* (Larose, 2005) is done. In it, with varying the values of input variables in the acceptable interval, the output variation is measured. With the interpretation of the output variation it is possible to recognize most influential input variable. To do it, at first the average value for each input variable is given to the model and the output of the model is measured. Then, *Sensitivity Analysis* for each variable is done separately.

For this purpose, the values of all variables except one in consideration are kept constant (their average value) and the model's response for minimum and maximum values of the variable in consideration are calculated. This process is repeated for all variables and then the variables with higher influence on variance of output are selected as most influential variables. For our problem, it means that the measure having most variation on output during analysis is the most important measure.

When one applies the above method on a category of measures, the most influential one is recognized. The selected measures from each category are then used to create a compound measure. Similar to the evaluation method, a table is constructed here too. As before, columns are the values of selected measures and an additional column records the target variable (0 or 1) showing the existence of a mapping between two entities. Now having such training samples a neural network is built. It is like a combined measure from the selected measures which can be used as a new measure for the extraction phase.

4.2 Experimental Results

In this section, results of the explained method are shown. Levenshtein (Levenshtein, 1966), NeedlemanWunsch (Needleman, 1970), SmithWaterMan (Smith, 1981), MongElkan (Monge, 1996), JaroWinkler (Jaro, 1995)(Winkler, 1999) and Stoilos (Stoilos, 2005) measures have been implemented using Jena API[2]. To be able to recognize mappings between entities with synonym names, a lexical measure which uses WordNet[3] is employed. In it, first a word is divided to its parts. For example, *bipedalPerson* is divided to *bipedal* and *person* terms. Then, using WordNet similarity of two words is calculated as follows:

$$ws(w_1,w_2) = \frac{|terms(w_1) \cap terms(w_2)|}{\max(|terms(w_1)|,|terms(w_2)|)} \quad (15)$$

Where ws stands for *WordNet Similarity*, terms is a function which get a word as input and return a set of the terms of that word as output, and ∩ is an operator which returns a set which contains terms which are synonym using WordNet.

EON_{2004} (Sure, 2004) data set is used for the Ontology Evaluation. From the tests in this collection tests numbered 203, 205, 222, 223, 230 are used to create initial train set necessary for our neural network model.

In this test, the reference ontology is compared with a modified one. Tests 204, 205, 221 and 223 are used from this group. Modifications involved naming conversions like replacing the labels with their synonyms as well as modifications in the hierarchy. We use these tests as a training set.

Also tests numbered 302, 303 and 304 are used as validation set. The reference ontology is compared with four real-life ontologies for bibliographic references found on the Web and left unchanged. We use tests 302, 303 and 304 from this group. This is the only group which contains real tests and may be the best one for evaluation of an alignment method.

After preparation of the train set table, *Sensitivity Analysis* as explained before is applied. Table 4 displays results of applying similarity analysis on each test set. In this table, second column shows the relative importance of measures used in the corresponding data set. As it is clear from the results, *Levenshtein similarity* is the most important one in predicting the relation of entities.

Table 4: Calculating Optimal Relationship Couple

Levenshtein Similarity	0.416
WordNet Similarity	0.415
Smith Waterman Similarity	0.023
Needleman Wunsch Similarity	0.011
Mong Elkan Similarity	0.010
Jaro Winkler Similarity	0.006
Stoilos Similarity	0.004

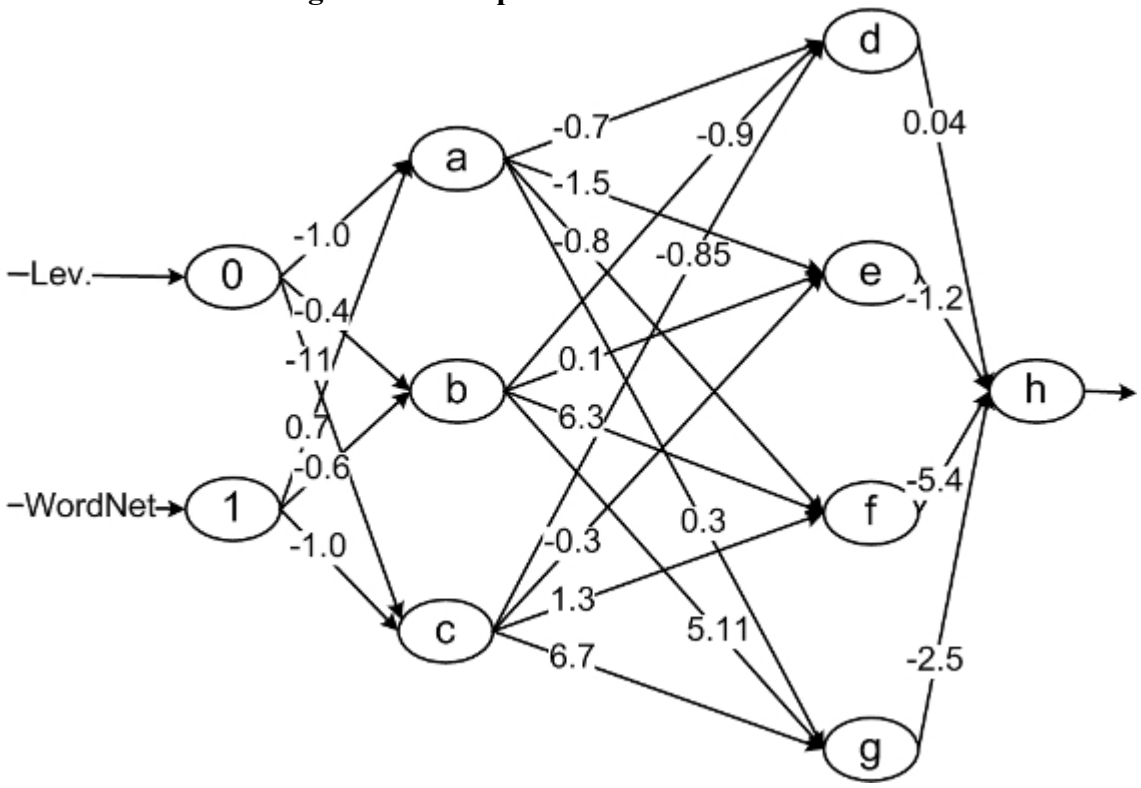
In the training phase five different models has been created explained hereafter. To obtain these models alpha=0.95, initial Eta=0.3 and Eta Decay=30 has been used.

- T.1- In this test, all the measures has been considered. To obtain a satisfactory model a dynamic approach to find a good value for number of layers and the number of neurons in the hidden layer is employed. As a result, a four layered model with

〈 7, 4, 5, 1 〉 neurons in input layer, two hidden layer and output layer, correspondingly, has been constructed.

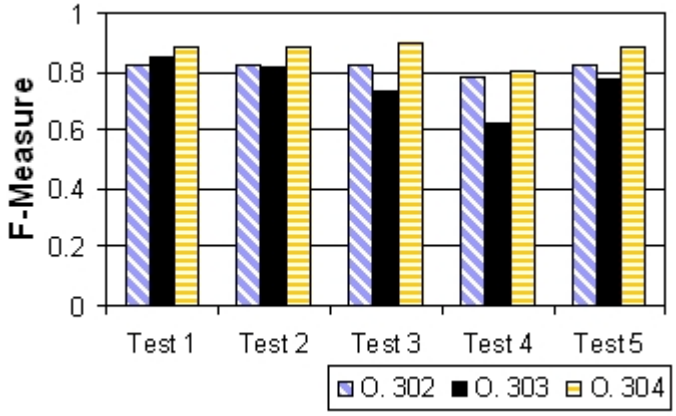
- T.2- In this test, *Levenshtein* and *WordNet* based measures which are selected from previous test is used. Here another four layer neural network with 〈 2, 3, 4, 1 〉 nodes is constructed as shown in Figure 1. According to this model and values obtained from *Levenshtein* and *WordNet* based methods by observing the output node, it is possible to decide if two entities are correspond.
- T.3- In this test, only *Stoilos* measure is used. The constructed model is in 〈 1, 2, 2, 1 〉 form.
- T.4- In this test, only *WordNet* measure is used and the constructed model is also in 〈 1, 2, 2, 1 〉 form.
- T.5- In this test, *Levenshtein* and *WordNet* measures has been used. The created model is in 〈 2, 40, 30, 1 〉 form.

Figure 6: A Simple Neural Network Model



The results of applying validation set on each of the models are shown in the Figure 7. In the figure *F-Measure* is the harmonic mean of Precision and Recall. Precision is the proportion of correctly recognized mappings to all the recognized mappings and Recall is the proportion of correctly recognized mappings to all the existed mappings. Also *Test 1 · Test 5* shows the models of *T. 1 · T. 5* as described. It should be noted that this results are obtained without any filtering or extraction operations. Applying such operations will results in higher precision since some unrelated mappings will be eliminated.

Figure 7: Results of Applying the Model On EON Data Set



As it is obvious from the figure, without using any customary heuristics and only using some simple linguistic measures, satisfactory results are obtained. In practice, we should use measures from other categories like structural or instance based measures which we expect to result in higher precision.

5. Conclusion

In this paper, we have developed an original measure for calculating structural similarity between entities of two ontologies, which is capable of recognizing more correspondences than can be recognized by current methods. The measure is a generalization of *Resnik* and *Similarity Distance* methods. Our proposed method behaves relatively stable against the *Granularity* heterogeneity, and this happens merely because of the special definition of *Reduced Relationship* in it. We have compared proposed method with some famous existing structural methods using EON_{2004} tests and results show higher *precision* and *recall*. Also in this paper, a new method for creation of compound measures is introduced, which is based on *Sensitivity Analysis* from *Data Mining* area. Evaluation results on this idea shows its effectiveness compared to other proposed approaches.

More works are needed to simplify our structural measure for actual applications. Also we intend to develop a complete framework in which our structural similarity measure, together with other measures can be used for real applications.

Acknowledgements

Authors wish to thank their colleagues in the Semantic Web laboratory of Sharif University of Technology for their valuable efforts and insights for this research.

References

- Bach, L., Dieng-Kuntz, R., & Gandon, F. (2004). On ontology matching problems (for building a corporate semantic web in a multi-communities organization). In: *Proceedings of the oICEIS*, 2004, 2004.
- Barthlemy, J.-P., & Gunoche, A. (1992). Trees and Proximity Representations. Chichester, West Sussex: John Wiley and Sons, 1992.
- Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., et al. (2005). Specification of a Common Framework for Characterizing Alignment. Knowledge Web, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, Tech. Rep. deliverable 2.2.1, February 2005.
- Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. (1984). *Classification and Regression Trees*. Belmont: Wadsworth, 1984.

- Dieng, R., & Hug, S. (1998). Comparison of personal ontologies represented through conceptual graphs. In: *Proceedings of the 13th ECAI Conference*, 1998, pp. 341-345.
- Doan, A., Domingos, P., & Halevy, A. (2003). Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning*, vol. 50, no. 3, pp. 279-301, 2003.
- Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2003). Learning to map ontologies on the semantic web. In: *Proceedings of the International World Wide Web Conference* (WWW), 2003, pp. 662-673.
- Ehrig, M., Staab, S., & Sure, Y. (2005). Bootstrapping ontology alignment methods with APFEL. In: *Proceedings of the 4th International Semantic Web Conference* (ISWC-2005), ser. Lecture Notes in Computer Science, Y. Gil, E. Motta, and R. Benjamins, Eds., 2005, pp. 186-200.
- Ehrig, M., & Sure, Y. (2004). Ontology mapping - an integrated approach. In: *Proceedings of the European Semantic Web Symposium* (ESWS), May 2004, pp. 76-91.
- Euzenat, J., Barrasa, J., Bouquet, P., Bo, J.D., et al. (2004). State of the Art on Ontology Alignment. Knowledge Web, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, deliverable 2.2.3, 2004.
- Euzenat, J., & Valtchev, P. (2003). An integrative proximity measure for ontology alignment. In: *Proceedings of Semantic Integration workshop at ISWC*, 2003.
- Euzenat, J., & Valtchev, P. (2004). Similarity-based ontology alignment in owllite. In: *Proceedings of ECAI*, 2004, pp. 333-337.
- Jaro, M. (1995). Probabilistic Linkage of Large Public Health Data Files. *Molecular Biology*, Vol. 14, pp. 491-498.
- Kalfoglou, Y., & Hu, B. (2005). Crosi mapping system (cms) results of the 2005 ontology alignment contest. In: *Proceedings of K-Cap'05 Integrating Ontologies workshop*, 2005, pp. 77-85.
- Larose, D.T. (2005). *Discovering Knowledge in Data*. New Jersey, USA: John Wiley and Sons, 2005.
- Levenshtein, V. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics-Doklady*, Vol. 10, pp. 707-710, August 1966.
- Madhavan, J., Bernstein, P., & Rahm, E. (2001). Schema matching using cupid. In: *Proceedings of the 27th VLDB Conference*, 2001, pp. 48-58.
- Maedche, A., & Zacharias, V. (2002). Clustering ontologybased metadata in the semantic web. In: *Proceedings of the 13th ECML and 6th PKDD*, 2002.
- Monge, A.E., & Elkan, C.P. (1996). The Field-Matching Problem: Algorithm and Applications. In: *Proceedings of the second international Conference on Knowledge Discovery and Data Mining*, 1996.
- Needleman, S., & Wunsch, C. (1970). A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins. *Molecular Biology*, Vol. 48, 1970.
- Noy, N., & Musen, M. (2001). Anchor-prompt: using non-local context for semantic matching. In: *Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence* (IJCAI), 2001, pp. 63-70.
- Ross, S. (1976). *A First Course in Probability*. Macmillan, 1976.
- Smith, T., & Waterman, M. (1981). Identification of Common Molecular Subsequences. *Molecular Biology*, 147(1), 195-197.
- Stoilos, G., Stamou, G., & Kollias, S. (2005). A String Metric for Ontology Alignment. In: *Proceedings of the ninth IEEE International Symposium on Wearable Computers*, October 2005, pp. 624-237.
- Sure, Y., Corcho, O., Euzenat, J., & Hughes, T. (2004). Eds. In: *Proceedings of the 3rd Evaluation of Ontology-based tools* (EON), 2004.

- Valtchev, P. (1999). *Construction automatique de taxonomies pour laide la reprsentation de connaissances par objets*. Ph.D. Dissertation, Universite Grenoble.
- Valtchev, P., & Euzenat, J. (1997). Eds., Dissimilarity measure for collections of objects and values, ser. Lecture Notes in *Computer Science*. London, UK: Springer, 1997, vol. 1280.
- Winkler, W.E. (1999). The State Record Linkage and Current Research Problems. U. S. Bureau of the Census, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, Tech. Rep., 1999.
- Zhong, J., Zhu, H., Li, Y., & Yu, Y. (1995). Using information content to evaluate semantic similarity in a taxonomy. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (IJCAI-95), 1995.
- Zhong, J., Zhu, H., Li, Y., & Yu, Y. (2002). Conceptual graph matching for semantic search. In: *Proceedings of Conceptual Structures: Integration and Interfaces* (ICCS-2002), 2002, pp. 92-106.

Footnotes:

[1] http://wordnet.princeton.edu

[2] Web Ontology Language

[3] http://jena.sourceforge.net

[4] Classification And Regression Trees

Bibliographic information of this paper for citing:

Abolhassani, H., Bagheri-Hariri, B., & Haeri, S.H. (2006). "On Ontology Alignment Experiments." *Webology*, **3**(3), Article 28. Available at: http://www.webology.org/2006/v3n3/a28.html

---
