Recovering Uncertain Mappings through Structural Validation and Aggregation with the MoTo System

Floriana Esposito
Computer Science Dept.
University of Bari, Italy
esposito@di.uniba.it

Nicola Fanizzi
Computer Science Dept.
University of Bari, Italy
fanizzi@di.uniba.it

Claudia d'Amato
Computer Science Dept.
University of Bari, Italy
claudia.damato@di.uniba.it

ABSTRACT

We present an automated ontology matching methodology, supported by various machine learning techniques, as implemented in the system MoTo. The methodology is two-tiered. On the first stage it uses a meta-learner to elicit certain mappings from those predicted by single matchers induced by a specific base-learner. Then, uncertain mappings are recovered passing through a validation process, followed by the aggregation of the individual predictions through linguistic quantifiers. Experiments on benchmark ontologies demonstrate the effectiveness of the methodology.

1. INTRODUCTION

Ontology matching [6] is among the most difficult tasks in the context of the *Semantic Web* (SW) research. Although a variety of automatic systems have been proposed so far, their performance may vary a lot depending on the different domains [3]. This problem is generally tackled by selecting the optimal matcher based on the nature of the matching task and the different features of the systems. This selection may involve Machine Learning techniques [10] for finding optimal configurations of the matchers, determining the appropriate heuristics / parameter values to achieve the best results [4].

We propose a comprehensive approach that differs from the previous ones for exploiting a combination of multiple matchers which are able to capture diverse aspects of the alignment. This should allow for overcoming the weakness of the individual matchers. The idea of ensemble learning is inducing multiple classifiers (matchers) so that the accuracy of their combination (different classifiers can complement and complete one another) may lead to a higher performance.

The proposed methodology is made up of two stages. In the first stage it uses individual base-learners and then a meta-learner to elicit certain mappings from those predicted by single matchers induced by the base-learners. This phase adopts the *stacking* [14], an ensemble learning technique, which seems the most appropriate for composing diverse learners. In the second stage, mappings that were previously deemed as uncertain can be recovered through a taxonomic / structural validation process, followed by the aggregation of the individual predictions made by linguistic quantifiers [7]. This stage may be considered of as a way of enriching the features utilized for the choice of the best matchers, in order to get a more effective combination [3].

The methodology is validated in a realistic setting on benchmark ontologies. In particular, we use datasets from past OAEI campaigns that provide a gold standard mapping as well as mappings created by different matching systems. Thus, our system can train a classifier on the outcome of different matching systems and learn what combination of results from different matchers with the best indication of a correct correspondence. This is competitive w.r.t. previous attempts of combining matchers which have often been based on *ad hoc* methods or had to be customized manually.

The paper proposes the following contributions: a hybrid approach for combining various matching systems using machine learning techniques and linguistic aggregation; a methodology is especially meant to recover cases of uncertain mappings: with structural validators adopting generalized similarity functions [11]; and an aggregation operator implementing a large choice of quantifiers [7]; 3) experiments on OAEI benchmark ontologies prove that recovering mappings (through validation and aggregation) can significantly improve the overall performance of the ontology matching system (especially in terms of recall).

2. STACKING AND AGGREGATION

Let $O. = \langle N_C^{\cdot}, N_R^{\cdot}, N_I^{\cdot} \rangle$ denote the input ontology, where $N_C^{\cdot}$, $N_R^{\cdot}$ and $N_I^{\cdot}$ stand, respectively, for the sets of the names for the concepts (classes), the roles (properties) and the individuals of the ontology. For simplicity, we will consider the problem of matching the concepts of some ontology $O_1 = \langle N_C^1, N_R^1, N_I^1 \rangle$ to those of $O_2 = \langle N_C^2, N_R^2, N_I^2 \rangle$. We will focus on the problem of finding equivalence mappings ($\equiv$), although the method could be extended to discover subsumption mappings ($\sqsupseteq$).

2.1 Application Context

The reference application context of our hybrid ontology system is illustrated by Fig. 1. Given two input ontologies under comparison O_1 and O_2, let us suppose the *matching system* eventually provides a *similarity matrix* for the entities belonging to either ontology. Namely each element of the matrix contains a value that indicates the similarity between the couple of entities related to the row and column. This may have been computed through any specific

SAC'10 March 22-26, 2010, Sierre, Switzerland.

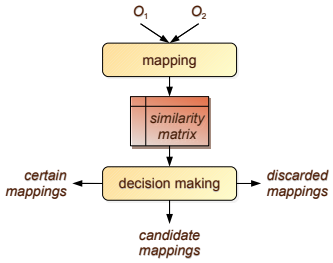

Figure 1: The reference application context.

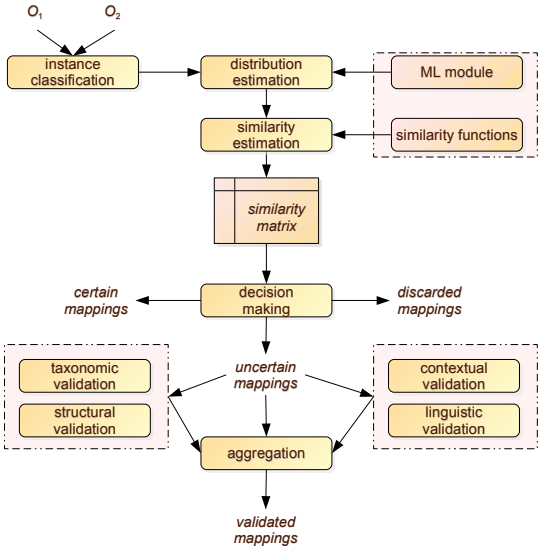

Figure 2: The MoTo system logical architecture.

technique which is aimed at determining such a value. On the ground of this matrix, a decision making module will tell the *certain* mappings (those whose similarity value exceeds a given threshold ε) from the others. If the system discarded all other mappings it would likely commit some errors, especially with those mappings that yielded a similarity which was not far from the threshold. Hence a number of uncertain mappings (i.e. a sub-matrix of the whole similarity matrix) are retained as *candidate* mappings to be evaluated along further techniques. To this purpose a *taxonomic validator* and a *structural validator* have been devised. They shall be discussed later in this section. Likewise, *aggregation operators* can be applied in the perspective of combining more validators. In such a perspective, the problem of assigning a degree of trust to each validator arises. In the following, we discuss our method for coping with this problem.

2.2 Functional Architecture

MoTo (*Mapping ontology To ontology*) is a multistrategy ontology mapping system. It resorts to a variety of matching techniques based on machine learning, linguistics and structural criteria. Additionally, it also implements aggregation principles for the mappings. Fig. 2 shows its functional architecture. The system is among the *composite* systems, according to the classification in [12, 6]. The main functions implemented as separate modules are:

- instance classification. All individuals from an ontology are collected and the membership of each of them w.r.t. each concept of the ontology is determined along with both a deductive and an inductive approach. This is performed through *deductive inference* performed with a standard OWL-DL reasoner and through *inductive inference* that is utilized when the response given by deductive reasoning is not satisfying, i.e. the reasoner cannot ascertain the membership (or non-membership) of an individual to some concept.

- distribution estimation. This module works on the ontologies to be matched utilizing the services of a separate ML module to estimate the probability distribution related to each couple of concepts $(C_1, C_2) \in N_C^1 \times N_C^2$. The ML module currently includes four *base-learners* and one *meta-learner* (implementing the technique of *stacking* of base-classifiers produced by base-learners) to combine their predictions [14]. The available base-learners produce a *k-Nearest Neighbor classifier*, an *Artificial Neural Network* and two *Bayesian Classifiers*, i.e. a *content* and a *name* classifier.

  This module is for estimating, for each couple (C_1, C_2), the joint probability of membership of the individuals to either concept. Each base-learner L receives a dataset made up of individuals from O_1 divided in the group of members of C_1 and the group of non-members of C_1. This is exploited to train a base-classifier. Then the individuals in O_2 are divided in two groups according to their membership to $C_2 \in N_C^2$. The classifier trained on the instances in O_1 is then used to classify the instances in O_2. Thus, four instance sets are formed, $U_2^{C_1,C_2}$, $U_2^{\neg C_1,C_2}$, $U_2^{C_1,\neg C_2}$ and $U_2^{\neg C_1,\neg C_2}$, corresponding to the combinations of inductive/deductive membership w.r.t. C_1 and C_2.

  The same procedure is repeated swapping the roles of the concepts (and related instances). The whole procedure can be iterated for each of the available base-learner. This is likely to lead to different outcomes, i.e. different U-sets. A simple procedure to make a decision in controversial classification cases may be based on a majority vote, which proves a sub-optimal but often quite effective a method [3]. In these cases *stacking* may come into service, with the meta-learner producing the final decision on the ground of the determination of the base-learners. The results provided by the ML-module are passed to the distribution estimator which computes the joint distributions related to the membership / non-membership to the two concepts.

- similarity estimation. This module receives the joint probability distributions computed by the previous one that are exploited when determining the concept similarity. A *similarity matrix* $S \in \mathbf{R}^{|N_C^1| \times |N_C^2|}$ is computed where each element $s_{ij} \in [0, 1]$ represents the similarity of the couple of concepts $(C_i, C_j) \in N_C^1 \times N_C^2$. Two thresholds $(\theta_{min}, \theta_{max})$ are determined to separate *certain* $\{(C_i, C_j) \in N_C^1 \times N_C^2 | s_{ij} \geq \theta_{max}\}$, *discarded* $\{(C_i, C_j) \in N_C^1 \times N_C^2 | s_{ij} \leq \theta_{min}\}$ and *uncertain mappings* $\{(C_i, C_j) \in N_C^1 \times N_C^2 | \theta_{min} \leq s_{ij} \leq \theta_{max}\}$. The first group can be output, the second one is simply discarded, while the last one is subject to further computation (*candidate mappings*).

- validation. Two types of validators were developed: *text-based validators* that use the vocabulary in conjunction with the ontological model (it will not be further detailed in the following) and *structure-based validators* that use the taxonomic structure of the input ontologies or also the entire graph established by the relationships (properties) therein.

- aggregation. the aggregation operator is activated at the end of the whole process for the composition of the results provided by the different validators to give a single decision regarding the uncertain mappings.

2.3 Taxonomic and Structural Validation

Following [5], five criteria have been considered for the comparison of two concepts:
#1 *most of their direct super-concepts are similar*;
#2 *most of their direct sub-concepts are similar;*
#3 *most of their super-concepts are similar*;
#4 *most of their sub-concepts are similar*;
#5 *all of their related concepts and properties are similar.*
Criteria #1-#4 merely employ the taxonomic relationship between concepts in the ontology. Validation following these criteria can be determined by a *taxonomic validator*. The final criterion considers also other relationships determined by the properties in the ontologies. This is implemented in a *structural validator*.

2.3.1 Taxonomic Validation.

The taxonomic validation module is based on the idea of comparing two concepts based on their respective position within the related subsumption hierarchy. The candidate mappings found to be uncertain in the previous phase are input to this validator to make a decision on their final rejection or a possible recover. By analyzing the taxonomies (parenthood relationships between the concepts) the module re-computes a similarity value for the candidate couples of concepts along the criteria #1-#4 above. This value is an estimate based on the similarity of their respective (direct) ancestors and/or (direct) descendants.

The taxonomic validation module requires an initial similarity matrix S_t containing the values that determined *certain mappings* and the couples of uncertain mappings that are to be validated. Each concept $C_i \in O_1$ is assigned with a unique concept $C_j \in O_2$ so that we have a injective mapping represented in S_t (which turns out to be sparse). The validator adopts an independent similarity measure σ. Fig. 3 depicts the algorithm for the n-th criterion. This algorithm takes into account both the observed average similarity and the rate of matches found. They are carefully combined since it should be considered that although the average similarity may be high the hierarchical structure of the ontologies should be also taken into account.

After computing the similarity of all couples of concepts from either ontology according to $\text{SIM}_i(C_1, C_2)$, $i = 1, \ldots, 4$, the final similarity value given by the validator is a linear combination of these values with an optional additional term, which stands for the similarity value computed via another function σ covering a different aspect.

2.3.2 Structural Validation.

The structural validator takes into account criterion #5 (see [5, 8]). The measure employed in this validator estimates the concept similarity through transitive relationships

```
SIM_n(C_1, C_2) : [0, 1]
input:  C_1, C_2: concepts;     // under comparison
        n: integer;     // criterion index
output: value: [0, 1];     // estimated similarity
begin
RC_1 ← set of concepts related to C_1 according to criterion n;
RC_2 ← set of concepts related to C_2 according to criterion n;
nMatches ← 0; sum ← 0; average ← 0; rate ← 0;
if max(|RC_1|, |RC_2|) = 0 then return 0;
else    for each (C_i, C_j) ∈ RC_1 × RC_2 do
                if S_t(C_i, C_j) > 0 then
                        sum ← sum + S_t(C_i, C_j);
                        nMatches ← nMatches + 1;
        if nMatches = 0 then return 0
        else    rate ← nMatches/ max(|RC_1|, |RC_2|);
                avg ← sum/nMatches;
return α · avg + β · perc;
end
```

Figure 3: Taxonomic similarity algorithm.

in the respective ontologies O_1 and O_2 and is based on the notion of *Information Content* (IC) [11].

We generalize the notion of *least common subsumer* (LCS) (or *closest common parent*) working on both ontologies. The structural validation module requires a similarity matrix S as input, containing the values that determined the *certain mappings* among the others. This matrix is sparse since each concept of O_1 is mapped to a single concept in O_2. A problem of *granularity* arises when an ontology may be more detailed than others w.r.t. a certain domain. LCSs can be extended with the *related concepts*: given (C_1, C_2), (RC_1, RC_2) is a couple of related concepts w.r.t. (C_1, C_2) and S iff

1. (RC_1, RC_2) are corresponding concepts in a mapping determined by S, $RC_i \in N_C^i$, and in the concept graph related to the ontologies there must exist a path from C_i to RC_i or viceversa, for $i = 1, 2$.

2. a list representing one such path (whose first element denotes the direction) is a *relation*; we will consider limited relations up to a certain *maxlen* with no loops.

3. there is at least a couple of *optimal* relations connecting C_1 and C_2 through RC_1 and RC_2. Optimality refers to a minimal total length (lensum) and *reduced relations*, i.e. those obtained by eliminating *isA* links and repeated transitive roles.

Given the set $RE(C_1, C_2)$ of the couples of related concepts w.r.t. the input ones, it is possible to define a structural similarity measure $\text{SIM}_s : N_C^1 \times N_C^2 \mapsto \mathbf{R}$ as follows

$$\text{SIM}_s(C_1, C_2) = w_0 \cdot \text{str}(C_1, C_2) + w_1 \cdot \text{map}(C_1, C_2)$$

where

$$\text{str}(C_1, C_2) = \sum_{(RC_1, RC_2) \in RE(C_1, C_2)} \frac{\sigma(RC_1, RC_2)^\alpha \cdot IC(RC_1, RC_2)}{\text{lensum}(RC_1, RC_2)^\beta}$$

with $\alpha, \beta \in [0, 1]$, $IC(C_i, C_j) = \sqrt{\log p(C_i) \cdot \log p(C_j)}$ and $\text{map}(C_1, C_2)$ is a similarity matrix computed on the ground of the *joint probability distributions* determined by the related module described before.

The IC function is generalized considering the estimated probability $\hat{p}(C) = \frac{\text{freq}(C) + M\mu}{|N_C| + M}$, where the *Laplace estimator* technique (for some $M \in \mathbf{N}$, $\mu \in [0, 1]$) is used to avoid values for which the function or its logarithm is undefined.

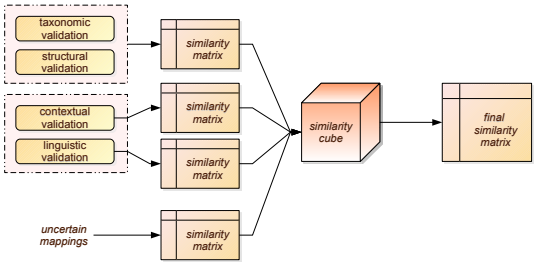

Figure 4: Aggregation scheme.

2.4 Aggregation

Similarity values computed according to different perspectives (matchers) ought to be aggregated to provide a final value. Various methods have been proposed to compute such aggregate values (see [2, 1, 13]). All require the computation of weights (confidence levels) for the different matchers implemented. The main problem is then how to determine these values automatically (e.g. by recurring to machine learning techniques). One possibility is the usage of an ad hoc ensemble learning technique like *stacking* [14]. The problem with these methods is the amounts of data required for training the individual and the meta-learner.

We resort to Yager's OWA (Ordered Weights Aggregation) operators [9]. Given a n-tuple of normalized weights $\vec{w}$, an *OWA operator* is a function $F : [0,1]^n \mapsto [0,1]$ such that: $F(a_1, \ldots, a_n) = \sum_{i=1}^{n} w_i b_i$ where the tuple $(b_1, \ldots, b_n)$ is obtained from $(a_1, \ldots, a_n)$ by sorting its element in descending order. Note that weights are associated to the positions rather than to the values.

The aggregation operator is based on weights computed through *linguistic quantifiers* [7]. A quantifier θ can be represented as $Q \subseteq I = [0,1]$, where for each $r \in I$, $Q(r)$ indicates the degree of satisfaction the criterion specified by θ. So if $\theta = most$ then $Q(.8) = 1$ means that 80% of the objects are compatible with this criterion. Suppose that n matchers produce n similarity values for the compared concepts to be aggregated $(\sigma_1, \ldots, \sigma_n)$, the value of the OWA operator is computed as before: $\theta(C_1, C_2) = F(a_1, \ldots, a_n) = \sum_{i=1}^{n} w_i b_i$. The weights are determined by the equations:

$$w_i = Q(i/n) - Q((i-1)/n) \qquad i = 1, \ldots, n$$

$$Q(r) = \begin{cases} 0 & r < a \\ (r-a)/(b-a) & a \leq r \leq b \\ 1 & r > b \end{cases} \qquad a, b, r \in [0,1]$$

where a and b are pre-defined thresholds for the single quantifiers. Nine quantifiers are implemented in MoTo. *Max* satisfy at least one matcher; *Min*: satisfy all matchers; *Avg*: identity – treats all similarity values equally; *Most*: satisfy most of the matchers; *Alh*: satisfy at least half of the matchers; *Amap*: satisfy as many as possible matchers; *Few*: satisfy few matchers; *NH*: satisfy nearly half of the matchers; *75Perc*: satisfy 75% few matchers (similar to *Most*).

The aggregation process consists of the following steps (see Fig. 4): 1) get the similarity matrices from the various validation modules; 2) reduce them to sparse ones such that each concept C_1 in one ontology corresponds to a single C_2 in the other; 3) create a *similarity cube* by composition of such matrices; 4) given a selected quantifier θ: compute the final similarity matrix based on the associated function

Table 1: Comparing the taxonomic validator (tv) to the base system (bs).

	a-n-c		k-a-c		k-a-n		k-n-c		k-a-n-c	
	bs	tv	bs	tv	bs	tv	bs	tv	bs	tv
204	.89	.91	.91	.91	.93	.93	.89	.91	.91	.91
205	.53	.57	.57	.61	.50	.52	.51	.58	.48	.57
206	.59	.64	.67	.74	.64	.67	.64	.74	.59	.64
221	.56	.56	.62	.62	.59	.59	.59	.59	.80	.80
222	1.0	1.0	.93	.91	.98	.97	.86	.89	.88	.90
223	.75	.87	.97	.97	.97	.97	.75	.87	.89	.91
228	.91	.99	.97	.99	.99	.99	.93	.99	.91	.94
233	.36	.36	.40	.40	.36	.36	.36	.36	.50	.50
239	.93	.92	.79	.89	.84	.90	.88	.87	.77	.81
240	.69	.84	.94	.96	.96	.96	.71	.94	.86	.90
301	.61	.61	.59	.61	.59	.61	.59	.61	.67	.69
304	.78	.82	.78	.82	.73	.75	.71	.78	.81	.81

3. EXPERIMENTATION

The experiments evaluated the improvement yielded by the adoption of the taxonomic and structural validators w.r.t. the performance of the base system. Moreover, we also wanted to find the related best linguistic quantifier.

Some of ontologies from the OAEI campaign[1] were selected. Specifically the suite was made up of a reference ontology (101) and a number of variants obtained by omitting properties, replacing names with synonyms, changing the graph structures, etc.. numbered 204, 205, 206, 221, 222, 223, 228, 233, 239, 240, 301, 304.

The four learners were set to their default parameters: Bayesian Name (n) and Bayesian Content (c) learners, k-Nearest Neighbor (k); Artificial Neural Network (a). Various combinations of the base-learners were possible. We will show the best ones, obtained by employing a choice of 3 or all four learners. For brevity, we omit the settings of all other parameters,

In Tab. 1 the results of the base matching system (bs) are compared to those of the taxonomic validation (tv), varying the choice of basic-learners. The table shows that tv improves w.r.t. bs in all but a couple of ontologies – namely, 221 and 233 – which have been obtained by eliminating the hierarchy. The improvement reaches the 15% for ontologies 223 and 240. Even more so, the improvement for 240 has been observed combining only 3 base-learners. This was probably due to the fact that the ontologies present an richer taxonomic structure (w.r.t. the original one). Conversely, for the same reason, slight decreases were observed for ontologies 222 and 239 because of their poorer taxonomic structures. In terms of precision and recall (omitted for brevity) the tv generally did not yield an improvement of the precision, as it may suggest erroneous mappings. Besides, the precision of bs with the ensemble of learners is already quite high, hence difficult to improve. The improvement is much more evident in terms of recall, as bs is not equally efficient.

Tab. 2 reports the outcomes of the experiments comparing the results of (bs) to those of the structural validation (sv) varying the choice of basic-learners. This table shows that sv improves the performance of the system, except for the ontology 233 for the same reasons outlined before. This was evident especially for ontologies 223, 239, and 240 (with a peak of +19% for 239) with the combination k-a-c. In particular, ontology 223 presents an extensive hierarchy which helped finding related concepts. A little decay was observed for ontology 221, one where the taxonomy was eliminated w.r.t. the original one, with opposite effects compared to the mentioned 223. Again, disaggregating these outcomes

[1] `http://oaei.ontologymatching.org`

Table 2: Comparing the structural validator (sv) to the base system (bs).

	a-n-c		k-a-c		k-a-n		k-n-c		k-a-n-c	
	bs	sv	bs	sv	bs	sv	bs	sv	bs	sv
204	.89	.91	.91	.93	.93	.94	.89	.91	.91	.93
205	.53	.58	.57	.56	.50	.52	.51	.57	.48	.49
206	.59	.64	.67	.69	.64	.67	.64	.64	.59	.62
221	.56	.55	.62	.60	.59	.58	.59	.58	.80	.82
222	1.0	1.0	.93	1.0	.98	1.0	.86	1.0	.88	.93
223	.75	.93	.97	.97	.97	.97	.75	.93	.89	.96
228	.91	.97	.97	.99	.99	.99	.93	.97	.91	.93
233	.36	.36	.40	.40	.36	.36	.36	.36	.50	.50
239	.93	.98	.79	.98	.84	.98	.88	.98	.77	.81
240	.69	.83	.94	.96	.96	.96	.71	.83	.86	.84
301	.61	.72	.59	.63	.59	.63	.59	.67	.67	.67
304	.78	.88	.78	.86	.73	.84	.71	.82	.81	.88

Table 3: Performance of the linguistic quantifiers.

Alh	Amap	Avg	Max	Min
.82	.75	.78	.85	.75
Most	**75Perc**	**Few**	**Nh**	**–**
.77	.78	.82	.81	-

in terms of precision and recall, one observers that precision of bs was already quite high and so difficult to be further improved: a single erroneously validated candidate mapping may even worsen the precision. Recall is improved by the validator w.r.t. the base system up to a 31% observed on ontology 239 (with the k-a-c combination) although the restriction of the hierarchical structure of this ontology w.r.t. the original one and the elimination of the properties might lead to predict this as a difficult case for sv.

Finally, we made experiments for testing bs together with the various additional components of the system, and especially the aggregation operator. Preliminarily, we tested the linguistic quantifiers. This produced a choice of the best quantifier to be utilized for the aggregation operator in the comparison with the other components (see Tab. 3). This has led to selecting the *Max* quantifier, although also *Alh*, *Few* and *Nh* produced good results.

Then experiments were performed comparing bs, tv, sv, linguistic-contextual validation (lcv) and aggregation (ao). Table. 4, reporting the average outcomes, shows that using ao the system was often able to produce better results. The most problematic ontologies were 222, 223, 239 and 301 for which ao lost some correct mappings. However, in general, ao produced the maximum average improvement w.r.t. the performance of bs. As with the previous experiments with the validators, the values for the precision index were difficult to improve. It is important to note that the choice of the *Max* quantifier sometimes led to erroneous mappings which diminished the overall performance of ao. The real gain is then in terms of recall with large improvements in some cases and only two cases were a minimal decay was observed (222 and 223) w.r.t. the performance of sv.

4. CONCLUSIONS AND OUTLOOK

This work focused on ontology matching validation based on structural aspects on the ontologies. It also concerned the aggregation of the similarities computed by different matchers, that are able to reconcile the various aspects targeted by each matcher through many linguistic quantifiers. The experimentation demonstrates that a combination of different techniques yields some added value in the quality of mappings found. Specifically, the taxonomic validator proved its effectiveness despite of the simplicity of the underlying idea. Another point in favor is that it can be applied to any kind of ontology being focused on the taxonomic aspect only. The structural validator goes a step even further as it exploits also other relationships between the concepts. The aggregation operator can select the best mappings from each system component and allows different types of aggregation by changing the most appropriate quantifier.

Table 4: Comparing the base system (bs) and the validators to the aggregation operator (ao).

	bs	lcv	tv	sv	ao
204	.91	.91	.91	.92	.92
205	.52	.56	.57	.54	.61
206	.63	.64	.68	.65	.69
221	.63	.81	.63	.63	.94
222	.93	.98	.93	.99	.97
223	.87	.95	.92	.95	.94
228	.94	.96	.98	.97	.98
233	.40	.52	.40	.40	.85
239	.84	.91	.88	.95	.89
240	.83	.86	.90	.88	.92
301	.61	.61	.63	.66	.64
304	.76	.83	.80	.86	.88

We are currently planning an enhancement of the validators so that other criteria may be implemented. The structural validator may also be enhanced by taking into account annotations/comments in natural language. A more in-deep investigation of the application of ensemble machine methods is also necessary. This may affect also the choice of weights for the aggregation operator.

5. REFERENCES

[1] D. Aumueller, H. Do, S. Massmann, and E. Rahm. Schema and ontology matching with coma++. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 906–908, 2005.

[2] A. Doan, P. Domingos, and A. Y. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *SIGMOD Conf.*, pages 509–520, 2001.

[3] K. Eckert, C. Meilicke, and H. Stuckenschmidt. Improving ontology matching using meta-level learning. In *Proc. of the Europ. Semantic Web Conf., ESWC2008*, volume 5444 of *LNCS*. Springer, 2009.

[4] M. Ehrig, S. Staab, and Y. Sure. Bootstrapping ontology alignment methods with apfel. In Y. Gil and et al., editors, *Proc. of the Int. Semantic Web Conf.*, volume 3729 of *LNCS*, pages 186–200. Springer, 2005.

[5] J. Euzenat and et al. State of the art on ontology alignment. Deliv. D2.2.3, Knowledge Web NoE, 2004.

[6] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.

[7] M. Grabisch, S. Orlovski, and R. Yager. Fuzzy aggregation of numerical preferences. *Fuzzy sets in decision analysis, operations research and statistics*, pages 31–68, 1999.

[8] B. Hariri, H. Abolhassani, and A. Khodaei. A new structural similarity measure for ontology alignment. In H. Arabnia, editor, *Proc. of SWWS 2006*, pages 36–42. CSREA Press, 2006.

[9] Q. Ji, P. Haase, and G. Qi. Combination of similarity measures in ontology matching using the OWA operator. In L. Magdalena and et al., editors, *Proc. of Information Processing and Management of Uncertainty in Knowledge-based Systems, IPMU08*, pages 243–250, 2008.

[10] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[11] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[12] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV:146–171, 2005.

[13] K. Tu and Y. Yu. CMC: Combining multiple schema-matching strategies based on credibility prediction. In L. Zhou and et al., editors, *Proc. of the Int. Conf. on Database Systems for Advanced Applications, DASFAA2005*, volume 3453 of *LNCS*, pages 888–893. Springer, 2005.

[14] D. Wolpert. Stacked generalization. *Neural networks*, 5:241–259, 1992.