Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods[⋆]

Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe

ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
{ifc|flav|cstroe1}@cs.uic.edu

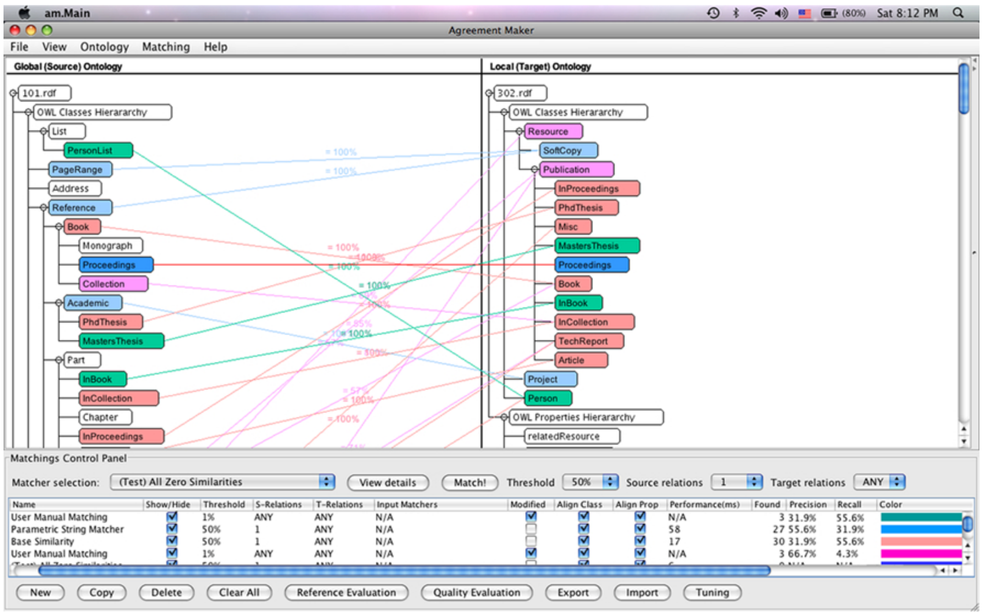
Abstract. The AgreementMaker system for ontology matching includes an extensible architecture that facilitates the integration and performance tuning of a variety of matching methods, an evaluation mechanism, which can make use of a reference matching or rely solely on "inherent" quality measures, and a multi-purpose user interface, which drives both the matching methods and the evaluation strategies. In this paper, we focus on two main features of AgreementMaker. The former is an optimized method that performs the selection of mappings given the similarities between entities computed by any matching algorithm, a threshold value, and the desired cardinalities of the mappings. Experiments show that our method is more efficient than the typically adopted combinatorial method. The latter is the evaluation framework, which includes three "inherent" quality measures that can be used both to evaluate matching methods when a reference matching is not available and to combine multiple matching results by defining the weighting scheme of a *fully automatic* combination method.

1 Introduction

The quest for correctness, completeness, and efficiency in the process of finding correspondences (or mappings) between semantically related entities of different real-world ontologies is a difficult and challenging task for several reasons. For example, an algorithm may be effective for a given scenario, but not for others. Even within the same scenario, the use of different parameters can change the outcome significantly. Therefore, state-of-the-art ontology matching systems [8] tend to adopt different strategies within the same infrastructure even though the intelligent combination of multiple matching results is still an open problem.

Our collaboration with domain experts in the geospatial domain [3] has revealed that they value automatic matching methods, especially for ontologies with thousands of concepts. However, they want to be able to evaluate the matching process, thus requiring to be directly involved in the loop. Such considerations have motivated the most recent features of the AgreementMaker system[1] for ontology matching [1, 2]. These features include a comprehensive user interface supporting both advanced visualization techniques and a control panel that drives all the matching methods and evaluation strategies (Figure 1) and an extensible architecture to incorporate new methods easily and to tune their performance. In this paper we concentrate on an optimization technique to produce the final set of mappings efficiently and on

⋆ Research supported by NSF Awards ITR IIS-0326284, IIS-0513553, and IIS-0812258.
[1] www.AgreementMaker.org.



Matcher selection: (Test) All Zero Similarities | View details | Match! | Threshold 50% | Source relations 1 | Target relations ANY

Name	Show/Hide	Threshold	S-Relations	T-Relations	Input Matchers	Modified	Align Class	Align Prop	Performance(ms)	Found	Precision	Recall	Color
User Manual Matching	✓	1%	ANY	ANY	N/A	✓	✓	✓	N/A	3	31.9%	55.6%	🟩
Parametric String Matcher	✓	50%	1	ANY	N/A		✓	✓	58	27	55.6%	31.9%	🟦
Base Similarity	✓	50%	1	ANY	N/A		✓	✓	17	30	31.9%	55.6%	🟥
User Manual Matching	✓	1%	ANY	ANY	N/A	✓	✓	✓	N/A	3	66.7%	4.3%	🟪

Fig. 1. User interface displaying side-by-side the source and target ontologies (top) and the control panel for the evaluation and comparison of matching methods (bottom).

the system's capability to evaluate, compare, and combine different strategies and matching results.

We describe next the main components of the paper. In Section 2, we cover related work. In Section 3, we describe several of the matching methods, or *matchers*, and their organization in *layers*. The ontologies being matched are called *source* and *target* ontologies. Matchers perform *similarity computation* in which each concept of the source ontology is compared with all the concepts of the target ontology, thus producing two similarity matrices (one for classes and one for properties), which contain a value for each pair of concepts.

In Section 4, we describe the process of *mappings selection* in which a similarity matrix is scanned to select the best mappings according to a given threshold and to the cardinality of the correspondences. For the mappings selection, we distinguish the following four cases: 1-1, n-m, n-$*$ (analogous to $*$-m), $*$-$*$, where 1, n, and m indicate specific input parameters and $*$ indicate that there is no constraint on the number of relations. For example, 1-1 means that each concept in the source ontology will be matched with at most one concept in the target ontology, n-m means that each concept in the source ontology will be matched with at most m concepts in the target ontology, whereas each concept in the target ontology will be matched with at most n concepts in the source ontology. In the case n-$*$, for example, each concept in the source ontology can be matched to any number of concepts in the target ontology. We note that in this case the chosen similarity threshold will in fact determine the number of concepts in the target ontology. In order to maximize the overall similarity of the selected mappings in a 1-1 or n-m matching, an optimization problem (namely the Assignment Problem) has to be solved. We provide an efficient solution to this problem by reducing it to the maximum weight matching in a bipartite graph and by adopting the Shortest Augmenting Path algorithm (SAP) [11]. Our experiments, which we describe in Section 6, have shown that this solution is considerably more efficient both space- and time-wise than the typically used Hungarian Method [12].

In Section 5, we describe the evaluation framework, which can make use of a reference matching or rely solely on “inherent” quality measures. In particular, we have adopted in our system two quality measures proposed by others [10], namely *order* and *distance preservation*, which analyze the structural properties of the produced matching to help determine its quality, and our own quality measure, called *local confidence*, which measures the reliability of the similarity measures assigned by a matching method. In addition, users can adopt any of these quality measures to define the weighting scheme of a fully automatic method that combines multiple matchings. The experiments, reported in Section 6, have shown that the *local confidence* quality measure can be quite effective in such a task.

2 Related Work

There are several notable systems related to ours [7, 8]. In this section, we will look at related systems with a special focus on the topics of combination of matching methods, mappings selection, and quality measures.

RiMOM [15] implements more than eight different matchers. It adopts a strategy selection method based on the definition of three ontology feature factors: *label similarity*, *structure similarity*, and *label meaning*. These factors are estimated based on the two ontologies to be matched. The matching strategies to be used are those that are suited to the highest factors. For example, if the two ontologies have high label similarity factor, then RiMOM will mostly rely on linguistic based strategies; while if the two ontologies have a high structure similarity factor, it will employ similarity-propagation based strategies on them. However, we note that the association between factors and strategies is predefined. Multiple results are combined using the weighted average of their similarity values, where the weights are predefined experimentally. While AgreementMaker does not provide a strategy selection method, it also provides a combination strategy based on the linear interpolation of the similarity values. However, in contrast with the RiMOM system, the weights can be either user assigned or evaluated through automatically-determined quality measures. This framework is extensible, because if a new method is integrated into the system, it can be directly used and combined with other methods. In terms of the final selection of mappings, RiMOM uses a similarity threshold value, while AgreementMaker uses in addition cardinality values.

Falcon-AO [9] uses four elementary matchers. Similarly to RiMOM, the association between detected similarities and matchers to be combined are predefined. However, matching results can only be combined two at a time (thus differing from both RiMOM and AgreementMaker). While RiMOM does not provide any evaluation strategy, Falcon-AO allows users to evaluate the precision, recall, and F-measure of a matching method given a reference matching. As for the mappings selection phase, Falcon-AO (like RiMOM) does not consider cardinality parameters.

SAMBO and SAMBOdtf [13] have five basic matchers, which are combined using the weighted average of similarities, where the weights are predefined. As for the mappings selection phase, SAMBOdtf adopts a strategy that is based on double threshold: pairs above the threshold are are retained as suggestions, those in between the lower and the upper threshold are filtered using structural information, and the rest is discarded.

None of the above systems proposes quality measures. One approach in this direction reduces mapping incoherence of the computed mappings to concept unsatisfiability in the ontology that results from merging matched ontologies [14]. The quality evaluation is then computed by measuring the effort necessary to remove all causes of incoherence from the matching.

Mapping incoherence is also used in the ILIADS system [16], which performs ontology matching and merging. They start by matching concepts, which are logical mappings that are used to create a unique integrated ontology. Logical reasoning over the constraints in the ontologies creates a consistent integrated ontology.

Other work proposes new measures that extend precision and recall to objects that are semantically defined, such as those in ontologies and alignments [5]. Such quality measures could be integrated into AgreementMaker, in addition to the "classically" defined concepts of precision and recall already supported.

3 Matching Methods

Our architecture allows for serial and parallel composition where, respectively, the output of one or more methods can be used as input to another one, or several methods can be used on the same input and then combined. A set of mappings may therefore be the result of a sequence of steps, called *layers*, to obtain a final *matching* or *alignment* (i.e., a set of mappings).

First layer matchers compare concept features (e.g., label, comments, annotations, and instances) and use a variety of methods including syntactic and lexical comparison algorithms as well as the use of a lexicon like WordNet in the Base Similarity Matcher (BSM) [4]. In the Parametric String-based Matcher (PSM) (see Figure 2), users can choose between a set of string comparison metrics (i.e., edit-distance, Jaro-Winkler, and a substring-based measure devised by us), define the normalization process (e.g., stemming, stop-word removing, and link stripping), and weigh the relevance of each considered concept feature. The similarity between two concepts is computed as the weighted average of the similarities between their single features.

In several methods, the common information between two concepts is kept into separate features and compared within each feature: labels are compared with labels and concept descriptions are compared with concept descriptions, for example. For this reason, we adopt a Vector-based Multi-word Matcher (VMM) that treats concepts as virtual documents containing the information pertaining to them. This information includes their descriptions, the information about their neighbors, and extensional information (e.g., class instances). These containers of terms are transformed into TF-IDF vectors and the similarity is computed using the cosine similarity metric, which is a common technique used to compare documents (see Figure 3).

Second layer matchers use structural properties of the ontologies. Our own methods include the Descendant's Similarity Inheritance (DSI) and the Sibling's Similarity Contribution (SSC) [4]. As their name indicates, they take respectively into account the information about concepts of which a given concept is a descendant or sibling.

Finally, third layer matchers combine the results of two or more matchers so as to obtain a unique final matching in two steps. In the first step, a similarity matrix is built for each pair of concepts, using our Linear Weighted Combination (LWC) matcher, which processes the weighted average for the different similarity results (see Figure 4). Weights can be assigned manually or automatically, the latter kind being determined using our evaluation methods (presented in Section 5). The second

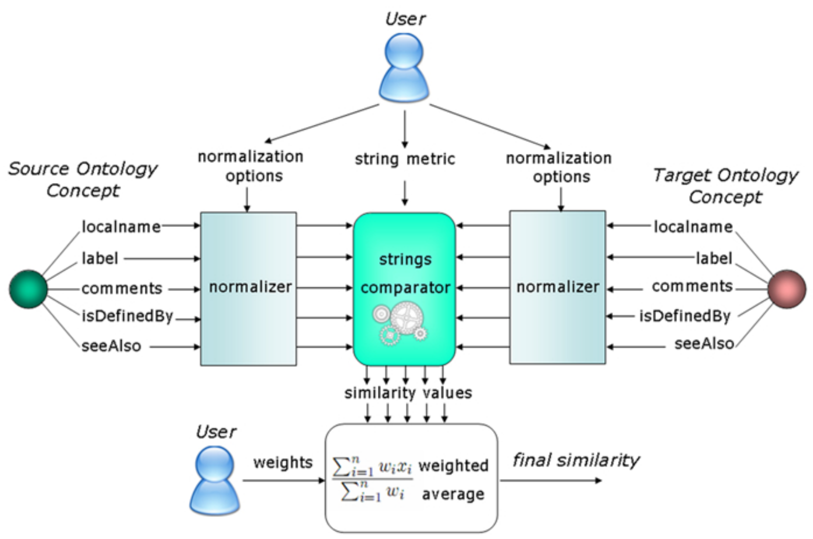


Fig. 2. Parametric String-based Matcher (PSM).

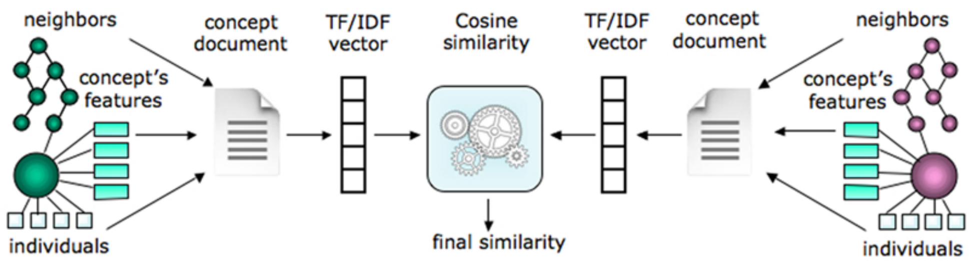


Fig. 3. Vector-based Multi-word Matcher (VMM).

step uses that similarity matrix and takes into account a similarity value and the desired cardinality to generate the final set of mappings, which maximizes the overall similarity while satisfying the selection constraints.

4 Mappings Selection

Four cases are considered in the selection process (see Section 1), depending on the desired cardinality: 1-1, n-m, n-$*$ (analogous to $*$-m), and $*$-$*$. The solution to n-$*$ can be found by scanning each row in the similarity matrix (or each column in the case $*$-m) and by selecting the n most similar correspondences with similarity values higher than the threshold (see Figure 5). For the $*$-$*$ case, only the threshold constraint has to be satisfied.

The 1-1 matching case, which is often required in real-world scenarios, is a challenging problem. In order to maximize the overall similarity in such scenarios, an optimization problem (namely the Assignment Problem) has to be solved. Usually, combinatorial algorithms (e.g., the Hungarian Method [12]) are used to find the optimal solution, but they are costly in terms of space usage and execution time and are for this reason impractical to match ontologies with thousands of concepts.

We provide an efficient alternative solution to this problem by reducing it to the maximum weight matching in the bipartite graph $G = (S \cup T, E)$, where S

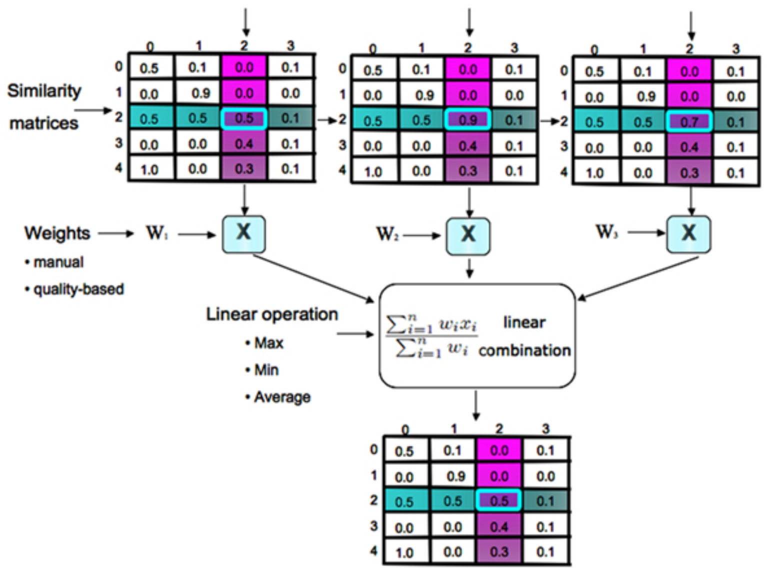


Fig. 4. Linear Weighted Combination (LWC) matcher.

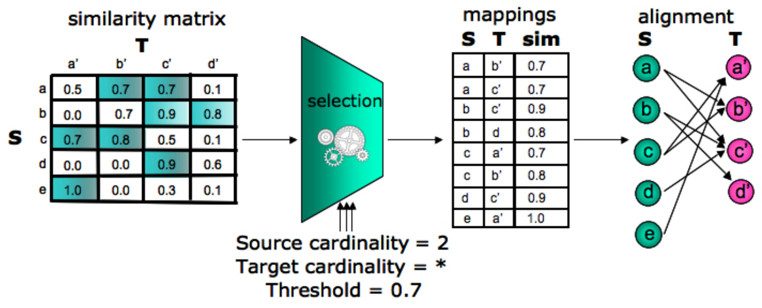


Fig. 5. Example of 2-$*$ matching with threshold 0.7 illustrating the similarity level (which yields the similarity matrix) and the selection level (which yields the alignment).

contains the source ontology concepts, T contains the target ontology concepts, and E contains an edge oriented from S to T for each correspondence with a similarity value higher than the threshold, weighted with the threshold value itself. We recall that a maximum weight matching M is a subset of the edges in E such that for each vertex in G at most one adjacent edge is contained in M and the sum of the weights (i.e., the similarity values) of the selected edges is maximized. Thanks to this transformation, we can adopt the Shortest Augmenting Path algorithm (SAP) [11] to find the optimal solution in polynomial time.

Finally, in the n-m selection case, we reuse our algorithm for the 1-1 matching case several times sequentially. We keep track of the number of mappings found for each vertex, and at the end of each iteration, we remove from the bipartite graph all the vertices together with their adjacent edges that have reached the maximal cardinality. The algorithm terminates when the graph is empty. We do not know of any other ontology matching system that has investigated the selection process with this level of detail.

	similarity level	**selection level**
local	Quality of each row (or column) of the similarity table	Quality of a mapping
global	Quality of the entire similarity table	Quality of the whole set of mappings

Table 1. Categorization of measures of quality of a matching method.

5 Evaluation

The most effective evaluation technique compares the mappings found by the system between the two ontologies with a reference matching or "gold standard," which is a complete set of correct mappings as built by domain experts, in order to measure precision, recall, and F-measure. The AgreementMaker system supports this evaluation technique. In addition, a reference matching can also be used to tune algorithms by using a feedback mechanism provided by a succession of runs.

However, a gold standard is usually not available. Therefore, "inherent" quality measures need to be considered. These measures can be defined at two levels as associated with the two main modules of a matcher: *similarity* or *selection* level. As illustrated in Figure 5, we can consider *local* quality as associated with a single row (or a single column) of the similarity matrix at the *similarity* level (or mapping at the *selection* level) or *global* quality as associated with all the correspondences in the similarity matrix at the *similarity* level (or with all the mappings in a matching at the *selection* level). This categorization of quality measures is summarized in Table 1. We have incorporated in our system two *global-selection* quality measures proposed by others [10] and one *local-similarity* quality measure that we have devised.

The intuition behind the two *global-selection* quality measures, namely (1) *order* and (2) *distance preservation*, is that given a set of mappings we can measure the structural properties of the produced matching to help determine its quality. In particular, according to (1), a matching should not change the order of concepts as defined by the *is-a* or *part-of* relations, and, according to (2), it should preserve the distance between concepts as much as possible. These metrics are good measures of the quality of a set of mappings if the ontologies are structurally similar.

In contrast with the two *global-selection* measures, the *local-similarity* quality measure is independent of the properties of the ontologies. Indeed, it tries to measure the reliability of the similarity measures assigned by a matching method, which is an intrinsic property of the matching method and therefore scenario independent. In particular, for each source (or target) concept we want to measure the confidence of the matcher as related to the selected mappings for that concept. Similarity-based matching techniques are based on the idea that if two concepts are very similar, they probably deserve to be matched. Therefore, our measure should be directly proportional to the similarity values of selected mappings. At the same time, we want to detect and penalize those matchers that tend to assign high similarity values too generously. For instance, if the correct solution is a 1-1 matching we expect each concept to be very similar (i.e., have high similarity value) to one concept at most, and very different (i.e., have low similarity value) to all others. Moreover, we want the similarity assignments to be stable in respect to the threshold value, so that changing the threshold slightly should not affect the final alignment considerably.

Therefore, given a matcher M and a concept c, we can define the *local confidence* of M with respect to c, $LC_M(c)$, as follows:

- let T be the set of all target concepts;

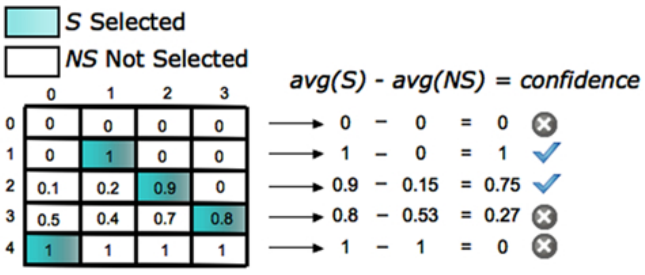


Fig. 6. Local confidence quality measure example.

– let $m_M(c) \subseteq T$ be the set of concepts $c' \in T$ that have been mapped to c by M;
– let $sim_M(c, c')$ be the similarity value between c and c' assigned by M;
– then $LC_M(c)$ is defined as the difference between the average of selected mappings' similarities for c and the average of the remaining correspondences' similarities:

$$LC_M(c) = \frac{\sum_{c' \in m_M(c)} sim_M(c, c')}{|\ m_M(c)\ |} - \frac{\sum_{c' \in (T - m_M(c))} sim_M(c, c')}{|\ T - m_M(c)\ |}.$$

With the reasonable assumption that M maps the most similar concepts, then $1 \geq \frac{\sum_{c' \in m_M(c)} sim_M(c, c')}{|m_M(c)|} \geq \frac{\sum_{c' \in (T - m_M(c))} sim_M(c, c')}{|T - m_M(c)|} \geq 0$, therefore $LC_M(c) \in [0, 1]$

A simple application of this quality measure is shown in Figure 6.

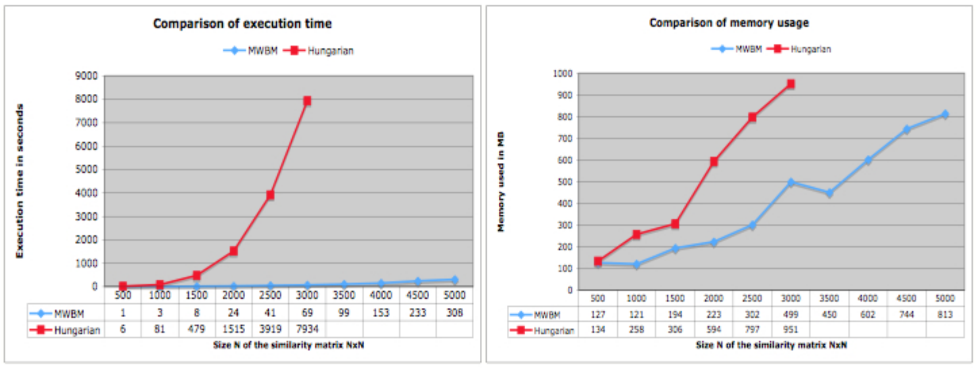
6 Experimental Results

In this section, we first report on the efficiency tests of the mappings selection algorithm. Then we compare the first layer matchers proposed in this paper (i.e., PSM and VMM) with the matching methods we used in the OAEI 2007 competition (i.e., BSM followed by DSI) [6]. Finally, we report on the results of the evaluation of the LWC matcher. We benefited from the capabilities of the AgreementMaker itself to perform the evaluations.

Mappings selection The most relevant module in this component is the 1-1 matching algorithm, which is also used to solve the *n-m* matching case. We compare the algorithm that we have adapted and implemented, the *Maximum Weight Bipartite Matching, MWBM,* with the Hungarian method [12] used by other matching systems.[2] In the first experiment, we ran both algorithms on random similarity matrices of different sizes (i.e., from a 500×500 matrix to a 5000×5000 matrix) with a threshold value of 0.5. As shown in Figure 7, the Hungarian method is much slower and uses a larger amount of memory.

In the second experiment, we investigated the effects of the threshold value on the performance of the algorithms. This time, we ran both methods on the same

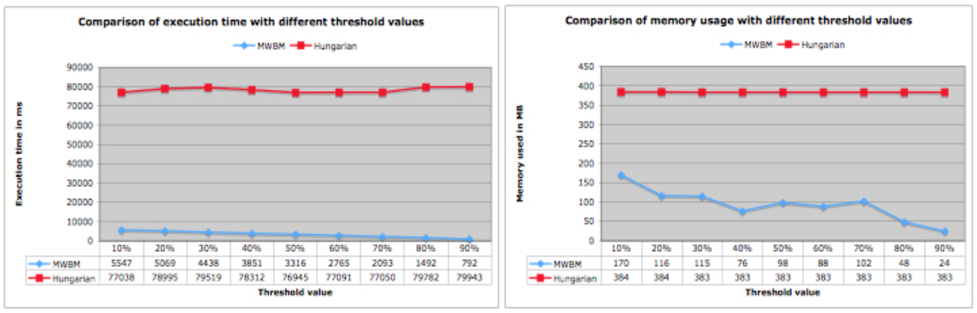
[2] The implementation is available at konstantinosnedas.com/dev/soft/munkres.htm.



	500	1000	1500	2000	2500	3000	3500	4000	4500	5000
MWBM	1	3	8	24	41	69	99	153	233	308
Hungarian	6	81	479	1515	3919	7934				

	500	1000	1500	2000	2500	3000	3500	4000	4500	5000
MWBM	127	121	194	223	302	499	450	602	744	813
Hungarian	134	258	306	594	797	951				

Fig. 7. Performance comparison between the Maximum Weight Bipartite Matching and the Hungarian method on different input sizes with a memory limitation of 1GB.



	10%	20%	30%	40%	50%	60%	70%	80%	90%
MWBM	5547	5069	4438	3851	3316	2765	2093	1492	792
Hungarian	77038	78995	79519	78312	76945	77091	77050	79782	79943

	10%	20%	30%	40%	50%	60%	70%	80%	90%
MWBM	170	116	115	76	98	88	102	48	24
Hungarian	384	384	383	383	383	383	383	383	383

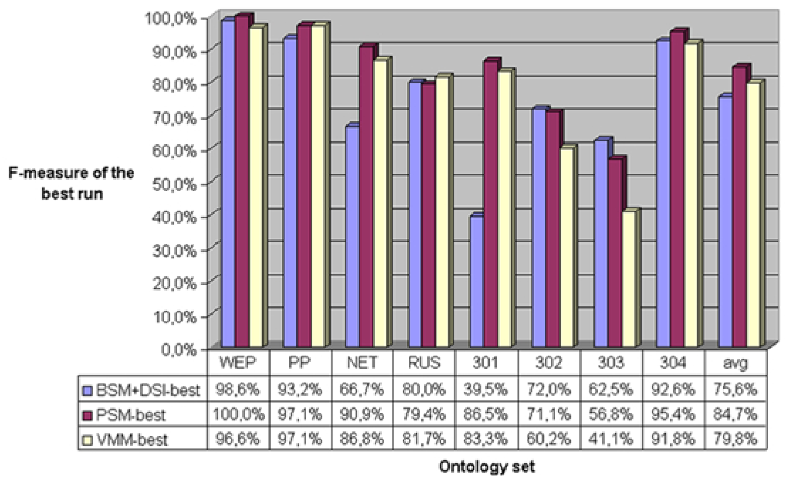
Fig. 8. Performance comparison between the Maximum Weight Bipartite Matching and the Hungarian method on different threshold values.

1000×1000 matrix using different threshold values (varying from 10% to 90%). As shown in Figure 8, the Hungarian method is not affected by the differences in the threshold values while the performance of MWBM improves when the threshold increases. That is, combinatorial matching methods, such as the Hungarian method, process the whole similarity matrix including those values that do not satisfy the threshold constraint. Instead, our algorithm transforms the similarity matrix into a weighted bipartite graph whose size is directly affected by the threshold value. Indeed, those correspondences that do not satisfy the threshold constraint are not translated into edges of the bipartite graph.

First layer matchers We ran the first two experiments on the alignment of eight pairs of ontologies. In particular, each set contains a source ontology, a target ontology, and the reference matching (expected matching) between them. The following ontology pairs were provided by I^3CON 2004:[3]

– **weapons set (WEP)** contains two classifications of various weapon types;
– **people and pets set (PP)**, contains two ontologies describing people and pets;
– **networks set (NET)** contains two classifications of computer networks;
– **Russia set (RUS)** contains general information about Russia.

[3] www.atl.external.lmco.com/projects/ontology/i3con.html



	WEP	PP	NET	RUS	301	302	303	304	avg
BSM+DSI-best	98,6%	93,2%	66,7%	80,0%	39,5%	72,0%	62,5%	92,6%	75,6%
PSM-best	100,0%	97,1%	90,9%	79,4%	86,5%	71,1%	56,8%	95,4%	84,7%
VMM-best	96,6%	97,1%	86,8%	81,7%	83,3%	60,2%	41,1%	91,8%	79,8%

Fig. 9. Comparison of first layer matchers (best run is considered).

The other four sets of ontologies are part of the OAEI benchmark.[4] The domain of these ontologies is bibliographic references. We consider those test cases in which the reference ontology *#101* has to be aligned with the following real-world ontologies:

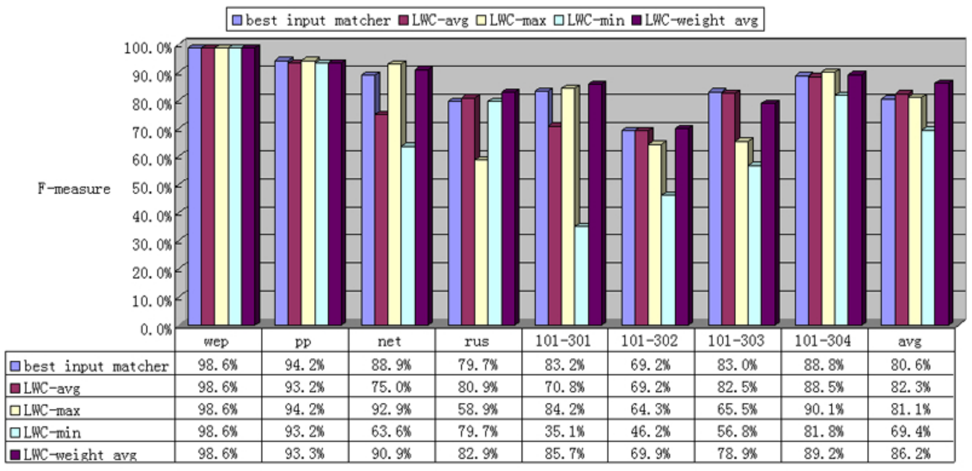
- **#301** is the BibTex bibliographic ontology from MIT;
- **#302** is the BibTex bibliographic ontology from UMBC;
- **#303** is the Karlsruhe bibliographic ontology used in the OntoWeb portal;
- **#304** is the INRIA bibliographic ontology.

In the first experiment, we ran the first layer matchers on all sets of ontologies using multiple threshold values for all of them. In Figure 9, for each method and for each ontology set, we report on the best F-measure of all runs. PSM is usually more effective than the others except for test cases #302 and #303, where it is slightly worse. However, the overall F-measure is definitely the highest. We further investigated the result of this experiment and noticed that BSM followed by DSI is quite accurate (high precision) but is able to find mappings only when the concepts are quite similar (otherwise displays low recall on dissimilar ontologies). Instead, PSM usually finds more mappings, even though some of them may be wrong occasionally. That is why it is less effective than BSM on the #303 set which contains mainly trivial mappings. VMM is sometimes better than the combination BSM+DSI, but it is usually worse than PSM. The problem is that these ontologies do not provide enough information to allow for this matcher to be very effective; however, it finds some non-trivial mappings not discovered by the other methods.

In summary, PSM is quite effective and stable, BSM is important for his high accuracy and VMM is able to find non-trivial mappings. Given the different qualities demonstrated by these matchers, we thought of combining them, thus motivating our next experiment.

LWC matcher We ran the first layer matchers (BSM, PSM, and VMM) and combined their results with the LWC matcher using four different linear operations: average of similarities, *LWC-avg*, maximum similarity, *LWC-max*, minimum similarity, *LWC-min*, and quality-based weighted average of similarities, *LWC-weight avg*.

[4] oaei.ontologymatching.org/2009/benchmarks/



	wep	pp	net	rus	101-301	101-302	101-303	101-304	avg
best input matcher	98.6%	94.2%	88.9%	79.7%	83.2%	69.2%	83.0%	88.8%	80.6%
LWC-avg	98.6%	93.2%	75.0%	80.9%	70.8%	69.2%	82.5%	88.5%	82.3%
LWC-max	98.6%	94.2%	92.9%	58.9%	84.2%	64.3%	65.5%	90.1%	81.1%
LWC-min	98.6%	93.2%	63.6%	79.7%	35.1%	46.2%	56.8%	81.8%	69.4%
LWC-weight avg	98.6%	93.3%	90.9%	82.9%	85.7%	69.9%	78.9%	89.2%	86.2%

Fig. 10. Comparison of combination techniques.

In the quality-based strategy, we adopted the *local confidence* quality to measure the relevance of the mappings generated by each matcher. In particular, the LWC method computes a combined similarity matrix that is obtained as the weighted average of the similarity matrices produced by the three matchers (see Figure 4). In this experiment, the weights are assigned by evaluating each matcher with the *local confidence* quality measure. Being a *local similarity* level quality measure (see Table 1), it defines a different value for each row of a similarity matrix, which is directly used to compute the weighted average for that row in the combination of the similarity matrices.

For each ontology set, we report in Figure 10 the best performance of the matchers to be combined, henceforth called *input matchers*, and the performance of the different versions of the LWC matcher. In most cases, almost all the mappings found by a single matcher are included in the set generated by another one, therefore any combination of these matchers cannot provide a significant improvement. A combined result equivalent to the best matcher in the input is already a good result. However, in all test cases, at least one of the combined results is equivalent to or better than the best result of the input matchers.

Considering the complexity of combining multiple matchings, which is still an open research problem, the most important result of this experiment is that the weighted average based on the local confidence quality is the most effective technique. Moreover, we note that the weights are chosen *automatically.*

7 Conclusions

In this paper, we make a contribution to the automatic evaluation of matchings by defining a quality measure that does not take into account the prior knowledge of a reference matching. We use this quality measure to define the weighting scheme of a *fully automatic* combination method. We also propose an efficient solution for the mappings selection task, whose performance is also positively affected by the threshold value. We plan to provide an API to make this functionality available to other matching systems.

In the future, we will take advantage of our extensible architecture and add new matching methods, for example, to our instance based methods. We plan to study new quality measures to enhance the current evaluation capabilities and our quality-based combination technique. Another direction for future research includes using partial reference matchings to perform the alignment of full ontologies.

References

1. I. F. Cruz, F. Palandri Antonelli, and C. Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.
2. I. F. Cruz, F. Palandri Antonelli, and C. Stroe. Integrated Ontology Matching and Evaluation. In *International Semantic Web Conference (Posters & Demos)*, 2009. To appear.
3. I. F. Cruz, A. Rajendran, W. Sunna, and N. Wiegand. Handling Semantic Heterogeneities using Declarative Agreements. In *ACM Symposium on Advances in Geographic Information Systems (ACM GIS)*, pages 168–174, 2002.
4. I. F. Cruz and W. Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications*, 12(6):683–711, December 2008.
5. J. Euzenat. Semantic Precision and Recall for Ontology Alignment Evaluation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, 2007.
6. J. Euzenat, A. Isaac, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, O. Šváb, V. Svátek, W. R. van Hage, and M. Yatskevich. Results of the Ontology Evaluation Initiative 2007. In *ISWC International Workshop on Ontology Matching (OM)*, volume 304, pages 96–132. CEUR-WS, 2007.
7. J. Euzenat, M. Mochol, P. Shvaiko, H. Stuckenschmidt, V. Svátek, W. R. van Hage, and M. Yatskevich. Results of the Ontology Alignment Evaluation Initiative. In *ISWC International Workshop on Ontology Matching (OM)*, volume 225, pages 73–95. CEUR-WS, 2006.
8. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.
9. N. Jian, W. Hu, G. Cheng, and Y. Qu. Falcon-AO: Aligning Ontologies with Falcon. In *K-CAP 2005 Workshop on Integrating Ontologies*, volume 156, pages 85–91. CEUR-WS, 2005.
10. C. Joslyn, A. Donaldson, and P. Paulson. Evaluating the Structural Quality of Semantic Hierarchy Alignments. In *International Semantic Web Conference (Posters & Demos)*, volume 401. CEUR-WS, 2008.
11. R. M. Karp. An Algorithm to Solve the $m \times n$ Assignment Problem in Expected Time $O(mn \log n)$. *Networks*, 10(2):143–152, 1980.
12. H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.
13. P. Lambrix, H. Tan, and Q. Liu. SAMBO and SAMBOdtf Results for the Ontology Alignment Evaluation Initiative 2008. In *ISWC International Workshop on Ontology Matching (OM)*, volume 431. CEUR-WS, 2008.
14. C. Meilicke and H. Stuckenschmidt. Incoherence as a Basis for Measuring the Quality of Ontology Mappings. In *ISWC International Workshop on Ontology Matching (OM)*, volume 431. CEUR-WS, 2008.
15. J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang. Using Bayesian Decision for Ontology Mapping. *Journal of Web Semantics*, 4(4):243–262, 2006.
16. O. Udrea, L. Getoor, and R. J. Miller. Leveraging Data and Structure in Ontology Integration. In *ACM SIGMOD International Conference on Management of Data*, pages 449–460, 2007.