2010 Workshops on Database and Expert Systems Applications

MeMO: A Clustering-based Approach for Merging Multiple Ontologies

Fabiana Freire de Araújo *Fernanda Lígia R. Lopes* *Bernadette Farias Lóscio*

Department of Computing
Federal University of Ceará
Fortaleza, CE, Brazil
e-mail: {fabiana, fernanda.ligia, bernafarias}@lia.ufc.br

***Abstract* — Since numerous ontologies are available on the Web, the requirement for merging such ontologies remains a pertinent issue in several applications. Many solutions were proposed in the literature to solve the ontology merging problem. However, these solutions just deal with the combination of two source ontologies at a time. To address the challenge of automatically merging multiple source ontologies, we propose a clustering-based approach. In our approach, named MeMO, the similarity among the source ontologies is calculated with the aim of defining the order in which they will be merged. A distinguishing point of our proposal is that we consider that better results are obtained when more similar ontologies are combined in the first place. We argue that the combination of ontologies with low level of similarity can introduce mistakes that will be carried out during the whole merging process. This argument is demonstrated in our experimental evaluation.**

Ontology merging; clustering techniques; Semantic Web

I. INTRODUCTION

The growing adoption of Semantic Web technologies has contributed significantly to increase the number of ontologies available on the Web. Since, in general, such artifacts are independently developed, it is very common to find several distinct ontologies, possibly overlapping, describing a given domain. Therefore, in this context, the creation of a global ontology, which integrates concepts from two or more source ontologies, can become a crucial task. Such task, called *ontology merging* [11], has been too much discussed in the recent years, and many solutions were proposed, like: iPROMPT [15], OMAlgorithm [17], COMA++ [5], ILIADS [21], Alignment API [10], HCONE-merge[14] and OntoMerge[9]. It is important to note that such solutions just deal with the combination of two ontologies at a time. To the best of our knowledge, none of the existing approaches for ontology merging considers the problem of automatically perform the merging of more than two ontologies.

However, there are situations where it can be necessary to deal with this problem, as, for example, in biological data integration applications [3]. In this sense, although the existing solutions can be extended for combining multiple ontologies, there are no guarantees about the quality of the obtained global ontology. In order to provide a "better" global ontology, the presence of a domain specialist is fundamental. In its absence, however, some extra device that approximates the final result to the expected one must be provided.

In this work, we propose an approach for automatic merging of multiple ontologies, called MeMO, which uses clustering techniques [1] in order to help the identification of the most similar ontologies. In our approach, we consider that when ontologies with a highest level of similarity are merged first, the resulting global ontology will be closest from the ideal one, which is the ontology that could be obtained if the merging process was manually executed by a domain specialist. In other words, we want to demonstrate that the similarity between ontologies can produce a positive impact on the final result of our merging process. Such idea is similar to the one proposed in [12] for multiple biological sequence alignment. To validate the proposed strategy, firstly we developed a prototype *(MeMOTool)* [4] and then several experiments were performed. Through such experiments, we have shown that better results are obtained when more similar ontologies are combined in the first place.

The remainder of this paper is organized as follows. Section 2 presents some relevant definitions about clustering techniques. These definitions are important to the understanding of the proposed ideas. Section 3 presents the MeMO strategy. Section 4 discusses the experiments. Section 5 presents an analysis of the experiments results. Finally, Section 6 contains the conclusions and some future work directions.

II. CLUSTERING TECHNIQUES

Clustering techniques are part of an area called Pattern Recognition [20], which aims to identify a certain pattern of an object of study for a future classification of other objects. Pattern recognition techniques can be classified in two categories: *supervised* and *unsupervised*. The main difference between them is that, in the first one, the characteristics of the objects of study are well known, including the category to which they belong. In the second approach, the category of the objects is unknown. In our work, we adopt the unsupervised approach, since there is no knowledge about the categories of data being classified. According to this approach, the main steps necessary for the discovery of groups from a set of objects of study are:

1529-4188/10 $26.00 © 2010 IEEE
DOI 10.1109/DEXA.2010.50

(i) *Definition of the clustering technique*: the first step consists in defining the clustering technique to be adopted [18]. In general, clustering techniques can be classified as: *hierarchical* and *non-hierarchical*. In the first one, a hierarchy (or a tree-like structure) is build in order to represent the relationship among entities (observations or individuals). In the non-hierarchical technique, a position in the measurement is taken as a central place and the distance is measured from such central point. Considering the nature of our problem, we adopt the hierarchical approach since this is the only technique that allows the combination of pairs of elements in order to compose a group. Other approaches use an attribute to compose groups of elements without really combining them.

(ii) *Definition of the similarity measure*: once the clustering technique is defined, the next step consists in choosing how to calculate the distance or similarity between the objects of study. The purpose of calculating the similarities or distances between objects is to create a matrix of similarity or distance, which represents the similarity among the considered objects. This matrix is the starting point of the hierarchical clustering technique.

(iii) *Definition of the clusters construction approach*: considering the hierarchical clustering technique, there are two methods that can be used for the clusters construction [1]: the *agglomerative* and the *divisive*. In the agglomerative method, initially, each object is considered as a single cluster (N objects correspond to N clusters) and in the following iterations, existing clusters are grouped together until there is just one single cluster. In order to combine existing clusters, the similarity measures among them must be computed. In the first iteration it is necessary to find similarities between all possible pairs of objects. In each of the following iterations, the similarity between the last cluster created and all other existing clusters must be computed. The divisive method works in an opposite way, i.e., it starts with a large cluster composed of all objects and such cluster will be subdivided into smaller groups until reaching a group with a single object. Considering the nature of our problem, we adopt the agglomerative method for clusters construction. Moreover, the agglomerative algorithms run in polynomial time while the divisive ones run in exponential time. Therefore, the divisive algorithms are the less used in the literature.

III. THE MEMO APPROACH

In this section we present our approach, named MeMO, for the automatic merging of multiple ontologies. Our approach is based on previous ideas proposed for multiple biological sequence alignment [12]. More specifically, we adopted the ideas related with the use of hierarchical techniques, where a final multiple sequence alignment is obtained through the progressive combination of pairs of alignments.

In a similar way, we adopted hierarchical clustering techniques to produce a global ontology from multiple source ontologies. Moreover, the MeMO approach produces a binary tree whose leaf nodes denote the source ontologies, and the root node represents the global ontology. The intermediary nodes represent the integrated ontologies, obtained during the merging process. One important aspect of MeMO is that it produces a global ontology which is really close to the ideal one.

The MeMO approach receives as input a set of ontologies and their corresponding ontology alignments. An ontology alignment is a set of correspondences between two ontologies and it is the result of the ontology matching process [11]. Such alignments can be obtained using existing ontology matching solutions, as for example *HMatch* [6] and *Alignment API* [10].

Following the ideas proposed by hierarchical clustering methods, two important tasks must be executed to produce the global ontology: the *similarity matrix building* and the *progressive ontology combination*, which are both explained below.

A. Similarity matrix building

The input for this task is composed by a set of source ontologies[1], $O = \{O_1,...,O_n\}$, and a set of alignments among them. The final result is a similarity matrix M_{nxn}, where each cell $M(i, j)$ denotes the similarity value between a pair of ontologies O_i and $O_j \in O$.

To calculate the similarity values, we adopted the ideas proposed by [19], where calculating the distance between two groups (clusters) corresponds to calculate the similarity between two ontologies; considering that the similarity corresponds to the distance between the groups and that an ontology corresponds to a group. Therefore, one of the existing approaches to calculate the similarity between two ontologies [19] can be used to populate the similarity matrix. More specifically, our approach adopts the function (1), called *DICE* (O_i, O_j), where: $|A(O_i, O_j)|$ denotes the number of correspondences of the alignment $A(O_i, O_j)$ and $|O_i|$ and $|O_j|$ denotes the number of elements of ontology O_i and O_j, respectively [8].

$$DICE(O_i, O_j) = \frac{|A(O_i, O_j)| + |A(O_j, O_i)|}{|O_i| + |O_j|} \quad (1)$$

After populating the similarity matrix with similarity values between all pairs of ontologies, the next task consists in progressively combining the source ontologies in order to obtain the global ontology.

B. Progressive ontology merging

The main goal of this task is to obtain the global ontology O_g, which represents an integrated view of the source ontologies $\{O_1,...,O_n\}$. During this task, is produced

[1] Each ontology O_i is composed by a set of entities $E_i = \{e_{i1}, ..., e_{im} \mid e_{ij} \in C_i \cup P_i$ and $1 \le j \le m\}$, such that C_i and P_i denotes, respectively, the set of classes and properties of O_i , $1 \le i \le n$.

a binary tree representing the progressive merging of the set of source ontologies. This task consists of two activities (*ontology combination* and *similarity matrix rebuilding*) which are iteratively repeated until obtaining a similarity matrix of order one (M_{1x1}).

Ontology combination

This activity starts with the analysis of the similarity matrix M in order to discover the pair of ontologies O_i and O_j with the highest degree of similarity. These ontologies are obtained from the identification of the cell $M(i, j)$ containing the largest score, in a given moment. Once the most similar ontologies were identified, the next step consists of applying the *merge* function in order to obtain a new ontology O_c, called *combined ontology*. The function $merge(O_i, O_j, A_w)$ receives as input two ontologies O_i and O_j, and the alignment $A_w = A(O_i, O_j)$ between O_i and O_j and it gives as a result a combined ontology O_c, such that $E_c = E_i \cup E_j$, $C_c \supseteq C_i \cup C_j$ and $P_c \supseteq P_i \cup P_j$. In this work, we considered the ideas of [16] in order to define the *merge* function, as well as to solve conflicts that may arise during the ontology merging process.

Similarity matrix rebuilding

Since a new ontology was produced in the last activity, in order to continue the proper execution of the progressive combination, a new similarity matrix must be produced. To produce this new similarity matrix, the first step consists in updating the existing matrix through the removal of the ontologies involved in the previous step, and the insertion of the new ontology O_c. Next, alignments between the combined ontology O_c and the remaining source ontologies, i.e, the ontologies that were not merged yet, must be identified. Such alignments are necessary to produce the similarity values between the new ontology and the remaining ones. Once the new alignments were identified, the new similarity values are produced, as described earlier. This activity finishes when the new similarity matrix is completely filled with the new similarity scores.

It is worth to mention that in order to minimize the human intervention during our merging process, we had to develop a new strategy for discovering ontology alignments [2]. In a general way, our solution is based on the following idea: when a new ontology O_c is generated from a combination of two other ontologies O_i e O_j, instead of performing a matching between O_c and the remaining ontologies of the matrix M, the new alignments can be obtained by *reusing* the alignments previously provided in the initial step of the multiple ontology merging process. This way, we propose the reuse of previous alignments to avoid the complexity of performing multiple ontology matching operations. The complexity of ontology matching process is discussed in [8]. Due to the lack of space, we do not present details about our alignments reusing process.

IV. EXPERIMENTS AND IMPLEMENTATION ISSUES

To validate the proposed approach, an open source prototype, called *MeMOTool*[2], was developed, and several experiments with a varying number of ontologies were performed. It is important to note that the *MeMOTool* can be used to perform the merging of multiple ontologies considering both, the MeMO strategy, as well as a random strategy. In our tests, we considered twenty (20) ontologies defined in OWL-DL language, extracted from Web sites like *Swoogle*[3] and *OntoSelect*[4]. The chosen ontologies describe concepts related to the academics domain and they have different number of classes (from 9 to 42), different number of properties (from 1 to 46) and different levels of depth (from 1 to 5). Table I describes the number of ontologies used in each experiment and its corresponding number of alignments. To identify the ontology alignments, we used the *Alignment API* [10], as well as the experience of a domain specialist.

TABLE I. DETAILS ABOUT THE PERFORMED EXPERIMENTS

	Number of Ontologies	Number of Alignments
Exp1	3	3
Exp2	4	6
Exp3	8	28
Exp4	12	66
Exp5	16	120
Exp6	20	190

We classified the ontologies used in our experiments in three categories:

- *Gold standard ontology (O_{Ref})*: represents the ideal ontology. Our Gold standard ontologies were created by domain specialists using iPROMPT tool [15];
- *MeMO global ontology* (O_{Comp}^{M}): represents a global ontology generated by the *MeMOTool*, considering the ontology similarity during the merging process;
- *Random global ontology* (O_{Comp}^{A}): represents a global ontology obtained as a result of multiple random executions of the merging process. These ontologies were generated using the *MeMOTool* but without considering the similarity measures.

Different sets of ontologies were created for each one of the performed experiments, including: one (1) gold standard ontology O_{Ref}, one (1) MeMO global ontology O_{Comp}^{M} and ten (10) random global ontologies O_{Comp}^{A}. In the last case, the random ontologies were created in order to calculate the average of the executions, considering that the obtained results can vary in each execution.

The measures *Precision, Recall* and *F-measure* [7] were used to compare these ontologies. In our analysis, we considered both the lexical and the structural ontology levels. In the lexical analysis, ontological concepts were

[2] http://www.lia.ufc.br/~fabiana/memo.html
[3] http://swoogle.umbc.edu/
[4] http://olp.dfki.de/ontoselect/

compared regarding just the exact string matching. Structural analysis focused on the comparison of hierarchies of concepts, i.e., the position of the ontological elements in a given hierarchy. More specifically, the following measures were considered in our experiments:

- *Lexical Precision (LP):* measures how many classes from the global ontology (MeMO or random) have the same name of a class belonging to the gold standard ontology;
- *Lexical Recall (LR)*: measures how many classes from the gold standard ontology are also present in the global ontology (MeMO or random) among the classes that should be in the global ontology, considering just the string matching;
- *Taxonomical Precision (TP):* measures how many elements present in the hierarchy of a concept in a global ontology (MeMO or random) are correct in relation to the elements present in the hierarchy of the corresponding concept in the gold standard ontology;
- *Taxonomical Recall (TR):* measures how many elements present in the hierarchy of a concept in the gold standard ontology are presented in the global ontology (MeMO or random) among the elements that should be there;
- *Taxonomic F-measure (TF) and Lexical F-measure (LF):* they are the harmonic-mean of precision and recall that take account of both measures.

To discover the values for the measures described above, we used the *OntEval*[5] tool and the following comparisons were performed: gold standard ontology with MeMO global ontology, and gold standard ontology with each one of the ten (10) random ontologies. All experiments were performed on a HP *Pavilion* Intel Core^{TM}2Duo (3GB main memory) running Linux Ubuntu 8.04.

V. EXPERIMENTS EVALUATION

Table II presents the results obtained in the series of experiments described in previous section. The results are organized in two main blocks. The first one, called MeMO, specifies the results of the comparison between O_{Ref} and $O_{Comp}{}^{M}$. The second block, called Random, specifies the results of the comparison between O_{Ref} and the ten (10) different versions of $O_{Comp}{}^{A}$. In our evaluation, we are mainly interested in the taxonomical measures values, i.e., TR, TP and TF. We made this choice because, when considering the whole set of experiments, such measures presented a higher variation in comparison with the lexical ones. Moreover, such measures are the most representative for the merging quality evaluation, since the focus of our approach is the comparison between the structure of the gold standard ontology and the resulting merged ontologies (MeMO and Random).

From Table II, we may observe that in the first experiment (*Exp1*), the MeMO strategy presented better results in comparison with the random strategy for TR and TF. In *Exp2*, with four (4) source ontologies, the difference becomes visible also for the TP measure. This difference is accentuated in *Exp3* and *Exp4* with eight (8) and twelve (12) source ontologies, respectively. One important observation is that these measures become far from the "ideal values" (TP and TR < 0.72) in the experiments *Exp5* and *Exp6*. Such "ideal values" were determined according to the experiments performed using the twenty (20) ontologies used in our tests.

TABLE II. EXPERIMENTS RESULTS

Strategy	Number of Ontologies	Number of Alignments	LP	LR	TP	TR	TF	Time (sec.)
Exp1								
MeMO	3	3	1.0	1.0	1.0	0.976	0.988	4.24
Random			1.0	1.0	1.0	0.915	0.954	3.9
Exp2								
MeMO	4	6	1.0	1.0	1.0	1.0	1.0	4.99
Random			1.0	1.0	0.95	0.92	0.97	4.5
Exp3								
MeMO	8	28	0.98	0.98	1.0	0.91	0.954	8.15
Random			0.98	0.98	0.97	0.94	0.931	8.2
Exp4								
MeMO	12	66	0.926	0.987	0.842	0.845	0.843	15.229
Random			0.907	0.964	0.703	0.773	0.735	16.4
Exp5								
MeMO	16	120	0.907	0.92	0.595	0.66	0.63	24.5
Random			0.897	0.952	0.622	0.714	0.66	24.16
Exp6								
MeMO	20	190	0.935	0.96	0.72	0.713	0.717	34.8
Random			0.91	0.94	0.69	0.701	0.697	31.72

It was observed that, from the values TP and TR< 0.7, the merged ontologies presented a considerable structural difference in relation to O_{Ref}. In fact, this is an empirical observation because what is a good result for one application cannot be for another.

Note that even when the values are far from 1.0, the MeMO strategy is still better than the random strategy. The only exception is the experiment *Exp5*, where the random execution achieved better results than the MeMO for some of the quality measures. In this experiment, the random strategy was closest from its better results. But, since this is a random strategy, it should be considered that different values could be obtained using the same source ontologies.

Fig. 1 presents a comparison between the values obtained for the TF measure in the six experiments. For the random approach, three values were chosen: the average of 10 values, the best case and the worst case.

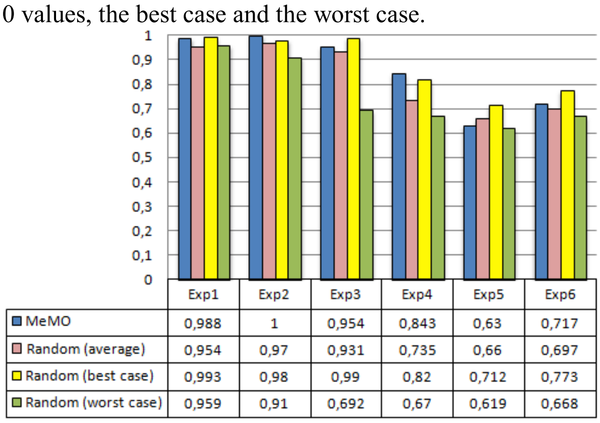

Figure 1. Comparative analysis based on F-measure values.

[5] http://nlp.shef.ac.uk/abraxas/index.html

Analyzing Figure 1, we can observe that the MeMO approach obtained better results than the average and also than the worst case of the random approach in all experiments, except *Exp5*. Moreover, the MeMO results were very reliable for the TF measure (0,954 to 1,0) regarding the merging of at most eight (8) ontologies (*Exp3*).

Another important observation is that when the number of source ontologies increases, the quality of the merging result decreases, considering both MeMO and random approaches. Moreover, the complexity of the source ontologies (e.g. number of classes, ontology depth) as well as the quality of the input alignments influence the final result. The MeMO strategy obtained excellent results until the merging of 12 (twelve) ontologies (*Exp4*). After experiments *Exp5* and *Exp6* (with 16 and 20 source ontologies, respectively) we may observe a greater number of inconsistent results for both strategies.

Consequently, we may see that the MeMO strategy is more stable and more reliable than a strategy where the similarity is not taken into consideration. MeMO was better in 20 of the 24 scenarios (see Table II and Figure 1). It is also important to analyze the absolute values differences between these two strategies. Although these differences are, in some cases, seemingly small, they are quite significant, because a small change in the structure of an ontology may cause an impact on the applications that use it. Besides, the best results of the random strategy are not the most frequents, therefore after many executions undesirable results can still be obtained. An interesting example can be seen in *Exp4*, where the worst case of the random strategy had the value of 0.692, while the strategy MeMO had the constant value of 0.954, representing a very large absolute difference.

VI. Conclusions And Future Work

This paper proposed an approach, named MeMO, for automatically merging multiple ontologies. The main distinguishing points of our approach are that the ontology similarity is considered during the merging, and that clustering techniques are used in order to group the most similar ontologies. Moreover, the MeMO approach always produces reliable results in contrast with a random approach, where it is not possible, for example, to assure that the best results will be discovered in a reasonable space of time. We validate our strategy through an experimental evaluation and even though the obtained results show that MeMO is promising, in some cases, the random strategy produces better results. In this sense, we intend to analyze such cases in order to derive improvements to our approach.

There are several directions for future work. First, we will test the limits of our approach, that is, how big is the ontology or the set of ontologies that we can support. Second, we want to study how to precisely determine the ideal values for the measures used in our experiments. Third, we may adopt the work presented in [13] to build our similarity matrix. Finally, regarding the formalization aspects, we intend to conduct a theoretical study to demonstrate that our algorithm achieves the desired results according to its conceptualization and implementation.

References

[1] M. R. Anderberg, "Cluster analysis for applications". New York: Academic Press, 1993.

[2] F. F. Araújo, "MeMO: Uma Estratégia para Merging Múltiplo de Ontologias baseada em Cluster. Master Dissertation. Federal University of Ceará (UFC), 2009.

[3] F. F. Araujo, A. M.A. Pinheiro, K.M. Farias, B.F. Loscio, and D. Oliveira, "FlagelLink A Decision Support System for Distributed Flagellar Data using Data Warehouse". In: Proceedings of The 23rd Annual ACM Symposium on Applied Computing (SAC), 2008.

[4] F. F. Araújo, H. R. Oliveira, F. L. R. Lopes, D. R. Cunha, D. Oliveira, and B. F. Lóscio, "MeMOTool: Uma Ferramenta para *Merging* Múltiplo de Ontologias". In: Proceedings of Demos Session, 24th Brazilian Symposium on Databases, 2009.

[5] D. Aumueller, H.H. Do, S. Massmann, and E. Rahm, "Schema and ontology matching with COMA++". SIGMOD Conference, 2005.

[6] S. Castano, A. Ferrara, D. Lorusso, and S. Montanelli, "The HMatch 2.0 Suite for Ontology Matchmaking". SWAP 2007.

[7] K. Dellschaft and S. Staab, "On how to perform a gold standard based evaluation of ontology learning". In: Proceedings of ISWC-2006 International Semantic Web Conference, 2006

[8] J. Euzenat and P. Valtchev, "An integrative proximity measure for ontology alignment". In:Proceedings of Semantic Integration workshop at ISWC, 2003.

[9] D. Dou, D. McDermott, and P. Qi, "Ontology translation on the semantic web". In: Journal on Data Semantics, II:35–57, 2005.

[10] J. Euzenat, "An API for Ontology Alignment". In: International Semantic Web Conference , 2004, pp. 698-712.

[11] J. Euzenat, and P. Shvaiko, Ontology Matching. Springer. Pg: 333. ISBN:3540496114, 2007.

[12] J.D Thompson, D.G Higgins, and T.J Gibson, "Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gag penalties and weight matrix choice". In: Nucleic Acids Res., v.22, 1994, p.4673-4680.

[13] Ismael Navas-Delgado, Maria del Mar Roldán-García, and José F. Aldana-Montes, "Semantic Fields: Finding Ontology Relationships". In: Proceedings of the 21st Dexa Conference, 2009.

[14] K. Kotis, G. Vouros, and K. Stergiou, "Towards automatic merging of domain ontologies: The HCONE-merge approach". In: Journal of Web Semantics, vol.4, no 1, 2006, pp. 60–79.

[15] N. F. Noy,and M.A. Musen, "The PROMPT suite: Interactive tools for ontology merging and mapping", Technical report, SMI, 2002.

[16] R. A. Pottinger and P.A. Bernstein, "Merging Models Based on Given Correspondences". In: Proceedings of the 29th VLDB Conference, Berlin, 2003.

[17] A. D. C. Rasgado and A. A Guzman, "A Language and Algorithm for Automatic Merging of Ontologies,". In: 15th International Conference on Computing (CIC'06), 2006, pp. 180-185.

[18] J. Sander, X. Qin, Z. Lu, N. Niu, and A. Kovarsky, "Automatic extraction of clusters from hierarchical clustering representations". In PAKDD - Pacific-Asia Knowledge Discovery and Data Mining, volume 2637 of LNAI, Springer Verlag, pp. 75–87, 2003.

[19] X. Su, X and J.A. Gulla, "An information retrieval approach to ontology mapping".Data Knowl. Eng. Vol. 58, no. 1, pp. 47-69, 2003.

[20] S. Theodoridis and K. Koutroumbas, "Pattern recognition". ISBN 0123695317, 2003.