A Dynamic Multistrategy Ontology Alignment Framework Based on Semantic Relationships using WordNet

Warith Eddine DJEDDI and Mohamed Tarek KHADIR,

Computer department, University Badji Mokhtar, LabGED, Annaba, Algeria
{djeddi,khadir}@labged.net

Abstract— **Ontology matching has emerged as a crucial step when information sources are being integrated. Hence, ontology matching has attracted considerable attention in both academia and industry. Clearly, as information sources grow rapidly, manual ontology matching becomes tedious, time-consuming and leads to errors and frustration. Thus the need for automated and semi-automated approaches becomes increasingly necessary and should especially consider the challenges of matching large schemas. This paper presents a dynamic multistrategy ontology alignment framework, named XMap++ (eXtensible Mapping) which exploits WordNet as a background knowledge sources. We propose a systematic approach to quantitatively estimate the similarity characteristics for each alignment task and propose a strategy selection method to automatically combine the matching strategies based on two estimated factors.**

Keywords-component; Ontology alignment; Linguistic similarity; Structural similarity; Dynamic strategies; Similarities aggregation; WordNet.

I. INTRODUCTION

Ontology alignment is a prerequisite in order to allow for interoperation between different ontologies and many alignment strategies have been proposed to facilitate the alignment task by (semi-) automatic means. Reference [1] provides an over-view of recent approaches including tuning frameworks such as Apfel and eTuner [2], [3]. Most previous approaches for automatic tuning apply supervised machine learning methods. They use previously solved match tasks as training to find effective choices for matcher selection and parameter settings such as similarity thresholds and weights to aggregate similarity values, e.g. [4]. A key problem of such approaches is the difficulty of collecting sufficient training data that may itself incur a substantial effort. A further problem is that even within a domain the successful configurations for one match problem do not guarantee sufficient match quality for different problems, especially for matching large schemas. Therefore, one would need methods to preselect suitable and sufficient training correspondences for a given match task, which is an open challenge.

The existing techniques are mostly based on calculating similarities between entities of two ontologies by utilizing various types of embedded information e.g., entity names, taxonomy structures, constraints, and entities' instances. These methods can be classified into two categories: the ones using a single strategy versus the others which combine multiple ones. In the former, all available information are defined as features in a single similarity function; while in the latter, different similarity functions are defined based on different types of information, and a composite method is used to combine the results of different similarities. In recent years, the combination method becomes more and more popular, due to its ease of extension and flexibility. In our previous work, we also proposed XMap++ [5] for ontology alignment by combining different strategies. Experimental results show that the combination method outperforms the single strategy based method in many cases.

The automatic alignment process suffers from the problem, where it is impossible, even for an expert knowledge engineer, to predict what entity alignment strategy is most successful for a given pair of ontologies. Furthermore, it is rather difficult to combine the multiple different sub strategies such as the individual similarity assessments (similar super-concepts, identical labels, same instantiations etc.) to behave optimally. This is especially the case with increasing complexity of ontology languages or increasing amounts of domain specific conventions.

A challenging issue in traditional methods is that both single and combination strategies are statically determined without considering characteristics of the alignment task.

Basically, we need an effective mechanism to automatically determine in what cases, a single strategy method should be used, and in what cases, a combination method should be used. Moreover, in a combination scheme, the lack of a systematic way to determine to what degree, each strategy should impact the alignment result remains an issue.

Based on these considerations, we extend our previous work [5] proposing dynamic multistrategy ontology alignment framework, which is still named XMap++.

The former version of XMap++ was a multiple strategy ontology alignment framework. It employs multiple ontology alignment strategies and sets the combination weight by manual. In the new version of XMAP++ proposed in this paper, given two input ontologies at runtime, it automatically determines, which ontology alignment methods to be used, what kinds of information to use in the similarity calculation and how to combine multiple methods as necessary. This paper aims at formalizing a dynamic multistrategy ontology alignment framework in an analytic and systematic way.

The rest of this paper is organized as follows: In Section 2, we give a formal definition of the ontology and ontology alignment, which formalizes the major tasks, in the dynamic multiple strategy ontology alignment. In Section 3, we give an overview of our framework XMap++. In Section 4, we describe the strategies alignment in XMap++. In Section 5, we give the experimental results. Finally, before concluding the paper, we discuss the possibilities for further improvements and future research.

II. ONTOLOGIES AND ONTOLOGY ALIGNMENT

A. *Ontology*

Definition 1: Ontology. An ontology is a formal specification of a shared conceptualization [6]. We describe the ontology as a 6-tuple: $\mathrm{O} = \{\mathrm{C}, \mathrm{P}, \mathrm{H}^{\mathrm{C}}, \mathrm{H}^{\mathrm{P}}, \mathrm{A}^{\mathrm{O}}, \mathrm{I}\}$, where C and P are the sets of concepts and properties, respectively. H^{C} defines the hierarchical relationships $\mathrm{H}^{\mathrm{C}} \subset \mathrm{C} \times \mathrm{C}$. $(\mathrm{c_i}, \mathrm{c_j}) \in \mathrm{H}^{\mathrm{C}}$ denotes that concept $\mathrm{c_i}$ is the subconcept of $\mathrm{c_j}$. Similarly, H^{P} defines the hierarchical relationships between each property and its subproperties, $\mathrm{H}^{\mathrm{P}} \subset \mathrm{P} \times \mathrm{P}$. A^{O} is a set of axioms. I is a set of instances of concepts and properties.

B. *Ontology Alignment*

Ontology alignment takes two ontologies as input and determines as the output the alignment result between entities of the input ontologies.

Definition 2: Ontology alignment. Given two ontologies O_1 and O_2, an alignment (or alignment task) finds, for each entity in O_1, a corresponding entity in O_2. O_1 is called the source ontology and O_2 the target ontology.

In this paper, we deal with principally with the task of ontology alignment; formally define an ontology alignment as the result:

$$\mathrm{Align}(\mathrm{O}_1, \mathrm{O}_2) = \left\{ \begin{matrix} (\mathrm{e}_{i1}, \mathrm{e}_{i2}, \mathrm{con}_i, \mathrm{relation}_i) | \, \mathrm{e}_{i1} \in \mathrm{O}_1, \mathrm{e}_{i2} \in \mathrm{O}_2, \mathrm{con}_i \in [0,1], \\ \mathrm{relation}_i \in \{\mathrm{exact}, \mathrm{narrower}, \mathrm{broader}, \mathrm{overlap}\} \end{matrix} \right\}. \quad (1)$$

$\mathrm{Align}(\mathrm{O}_1, \mathrm{O}_2)$ represents that entity e_{i1} in O_1 is aligned to entity e_{i2} in O_2 with the confidence con_i and the alignment type $\mathrm{relation}_i$.

III. PROPOSED ALGORITHM

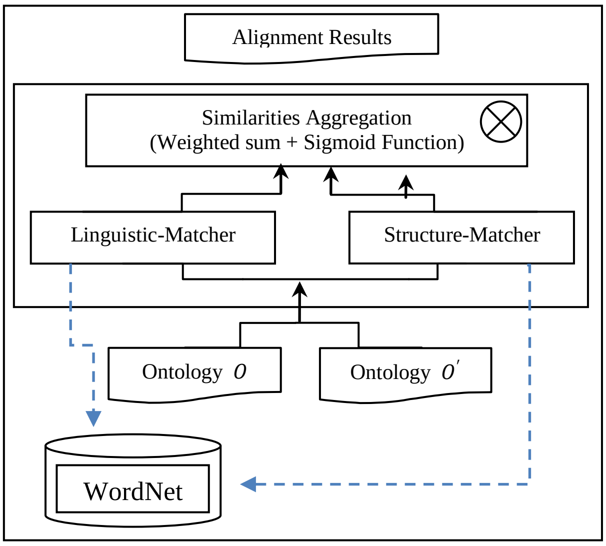

Figure 1. XMap++ architecture

We perform both lexical and structural comparisons in order to determine if concepts in different ontologies (OWL-DL) should be considered semantically compatible. We use a refinement approach, broken into three successive steps, illustrated in Figure 1.

Our strategy proposes to use structural comparison, where concepts that were once identified as lexically equivalent are now structurally investigated.

Making use of the intrinsic structure of ontologies, a hierarchy of concepts connected by subsumption (subClassOf) or equivalence (equivalentClass) relationships [7], we now isolate and compare concept sub-trees. Investigation on the ancestors (super-concepts) and descendants (subconcepts) will provide the necessary additional information needed to verify whether the pair of lexically equivalent concepts can actually be assumed to be semantically compatible.

A. *First Step: Lexical Comparison*

The goal of this step is to identify lexically equivalent concepts. Each concept label in the first ontology is compared to every concept label present in the second one, using lexical similarity as the criteria. Besides using the label itself, synonyms are also used, referring to the semantic relationships derived from the lexical system WordNet [8]. The use of synonyms enriches the comparison process because it provides more refined information. As a result of the first stage of the proposed strategy, the original ontologies are enriched with links that relate concepts identified as lexically equivalent.

B. *Second Step: Structural Comparison*

Comparison is done at this stage, and is based on the subsumption relationship that holds among ontology concepts. Ontology properties and restrictions are into consideration.

C. *Third Step: Fine Adjustments based on Similarity Measurements*

The third and last step is based on similarity measurements. Concepts are rated as very similar or little similar, based on pre-defined similarity thresholds and dynamic calculation of linguistic weight. We only align concepts that were both classified as lexically equivalent in the second step, and thus rated very similar. Thus the similarity measurement is the deciding factor responsible for fine tuning our strategy.

IV. Ontology Alignment Strategies In XMap++

A. *Similarity Factors between two Ontologies*

To capture the meaning of names in an ontology during the matching process, it is often referred to a thesaurus of terms and terminological relationships among them. The thesaurus is automatically derived from the lexical system WordNet [8]. WordNet is a well-known online lexical reference knowledge base and contains the semantic relationships from synset, a set of synonyms representing a distinct concept. Synsets provide different inter relationships such as synonymy and antonymy, hypernymy and hyponymy (superconcept and subconcept), meronymy and holonymy (Part-Of and Has-a). A query interface allows a user to search a term t in the database and returns a WordNet definition in natural language, and its generalizations, its specializations and under which it is bound by a relationship of composition to the different meanings of this term (the different synsets to which it belongs).

WordNet can be used in different ways to search for mappings. The first technique is explained in [9] and consists in extending systematically the label of a concept synonymous with belonging to synset the label of each term in WordNet, which allows, for example, approximating "Person" to "Human". It is based on psycholinguistic theories to define word meaning and model not only word meaning associations but also meaning-meaning associations [10].

For two ontologies O_1 and O_2, we define two similarity metrics, label similarity factor $\mathrm{Sim}_{\mathrm{LA}(O_1,O_2)}$ and structure similarity factor $\mathrm{Sim}_{\mathrm{Str}(O_1,O_2)}$, their values range from 0 to 1.

Definition 3: Linguistic features. Linguistic features refer to names of ontology elements and their meaning:

$$Sim_{LA(O_1,O_2)} = \frac{\mathrm{S}_{\mathrm{ConceptLA}}(t_c, t_{c'}) + \mathrm{S}_{\mathrm{PropertyLA}}(t_p, t_{p'})}{\max(|C_1|+|P_1|, |C_2|+|P_2|)}, \quad (2)$$

where |C1| and |C2|, |P1| and |P2| represent the number of concepts and the number of properties in O_1 and O_2, respectively.

The aim of the term affinity functions $\mathrm{S}_{\mathrm{ConceptLA}}(t_c, t_{c'})$ and $\mathrm{S}_{\mathrm{PropertyLA}}(t_p, t_{p'})$ is to evaluate the affinity between two terms t and t′ with respect to Word-Net.

$\mathrm{S}_{\mathrm{ConceptLA}}(t_c, t_{c'})$ of two terms t and t′ is equal to the value of the highest-strength path of terminological relationships between them in Word-Net if at least one path exists. Otherwise, it relies more on the value of the function $\boldsymbol{Similarity_{Name}}(t_c, t_{c'})$ [5]. Path strength is computed by multiplying the weights associated with each terminological relationship involved in the path, that is:

$$\mathrm{S}_{\mathrm{ConceptLA}}(t_c, t_{c'}) = \left\{ \begin{array}{c} \max_{i=1..k} \{w_{t_c \to_i^n t_{c'}}\} \text{ if } k \geq 1 \\ |\ Similarity_{\mathrm{Name}}(t_c, t_{c'})|\ \ \textbf{Otherwise} \end{array} \right\}, \quad (3)$$

$$\mathrm{S}_{\mathrm{PropertyLA}}(t_p, t_{p'}) = \left\{ \begin{array}{c} \max_{i=1..k} \{w_{t_p \to_i^n t_{p'}}\} \text{ if } k \geq 1 \\ |\ Similarity_{\mathrm{Name}}(t_p, t_{p'})|\ \ \textbf{Otherwise} \end{array} \right\}, \quad (4)$$

where: k is the number of paths between t and t′ in WordNet; $w_{t \to_i^n t'}$ denotes the *ith* path of length $n \geq 1$; $w_{t \to_i^n t'} = w_{1_{tr}} . w_{2_{tr}} \ldots . w_{n_{tr}}$ is the weight associated with the *ith* path, where $w_{j_{tr}} \mid j = 1,2,\ldots,n$ denotes the weight associated with the *jth* terminological relationship in the path.

The calculation of $\mathrm{Similarity}_{\mathrm{Name}}(t_c, t_{c'})$ and $\mathrm{Similarity}_{\mathrm{Name}}(t_p, t_{p'})$ can refer to our paper [5].

Definition 4: Structure similarity factor. The structure similarity factor evaluates the similarity of two ontologies based on their structure information:

$$Sim_{Str(O_1,O_2)} = Similarity_{str}(D_{context}(c), D_{context}(c')) . \quad (5)$$

We note by C the set of classes for a given ontology. The calculation of $\mathrm{Similarity}_{\mathrm{str}}(\mathrm{D}_{\mathrm{context}}(c), \mathrm{D}_{\mathrm{context}}(c'))$ can refer to our paper [5].

B. *The final Similarity Calculation*

The similarities reported by the different strategies of ontology alignment are combined as follows:

$$sim(e_1, e_2) = \frac{(w_{LA}.\sigma(sim_{LA(e_1,e_2)}) + w_{Str}.\sigma(Sim_{Str(e_1,e_2)})}{(w_{LA} + w_{Str})}, \quad (6)$$

where $sim(e_1, e_2)$ is the similarity of the combined entity e_1 et e_2. The weights w_{LA} and w_{Str} are the weights of different strategies. σ is a function of the sigmoid,

$$\sigma(x) = \frac{1}{(1+\exp(-5(x-\alpha)))}, \quad (7)$$

where α set to 0.5, the weights w_{LA} et w_{Str} are determined by

$$w_{LA} = \frac{Sim_{LA}}{\max(Sim_{LA}, Sim_{Str})}, \quad (8)$$

$$w_{str} = \frac{Sim_{Str}}{max\ (Sim_{LA}, Sim_{Str})}\ . \qquad (9)$$

When Sim_{LA} is bigger than Sim_{Str}, the combination relies more on the lexical similarity. Otherwise, it relies more on the resemblance which is interested to the external and internal structure of the entities.

Remark 1. In earlier versions of XMap++, the w_{LA} is defined by the user before the alignment of two ontologies. In this report we have defined a dynamic strategy for automating the choice of the linguistic weight. The originality of our approach is to consider as soon as possible in the context of alignment using the weighted sum in addition to the sigmoid function whose value changes according to the weight of the linguistic measure (w_{LA} : calculated automatically). The calculation becomes richer, more complete and accurate in the choice of the weight value w_{LA}.

The procedure is given by the following algorithm:

Algorithm $XMAP(c, c', \omega_{LA})$

Input the two concepts c and c', and the weight ω_{LA}
Output final similarity between c and c'
Begin algorithm
Definition n and n' are the names of c and c', respectively;
Definition $D_{context}(c) = [\,]$, $D_{context}(c') = [\,]$ the vectors of properties description of c and c', respectively;
Definition $portion - dsp = [\,]$ as one pair of the form (np,tr), of which np is the associated name to a property and tr $\in$ {OWLSomeValuesFrom, OWLMinCardinality>=1, OWLCardinality>=1, other restrictions}
Definition $x = 0,\ y = 0,\ Similarity_{final} = 0;$
$P(c) = \{p_i | (p_i, c, tr)\}$;
$x = Sim_{LA}(c, c')$;
For each property $p(c) \in P(c)$
/* $tr(p_i, c)$ the restriction type of p_i which $\in c$
$portion - dsp = [p(c), tr(p_i, c)]$;
Add $portion - dsp$ in $D_{context}(c)$;
For each property $p(c') \in P(c')$
/* $tr(p_i, c')$ the restriction type of p_i which $\in c'$
$portion - dsp = [p(c'), tr(p_i, c')]$;
Add $portion - dsp$ in $D_{context}(c')$;
$y = Sim_{Str}(D_{context}(c), D_{context}(c'))$;
$w_{LA} = \frac{x}{max\ (x,y)}$; /* calculated automatically
if $\omega_{LA} > 0.7$ then
$Similarity_{final}(c, c') = \omega_{LA} \cdot x + (1 - \omega_{LA}) \cdot sigmoid(y - 0.5)$;
else if $\omega_{LA} < 0.4$ then
$Similarity_{final}(c, c') = \omega_{LA} \cdot sigmoid(x - 0.5) + (1 - \omega_{LA}) \cdot y$;
else
$Similarity_{final}(c, c') = \omega_{LA} \cdot x + (1 - \omega_{LA}) \cdot y$;
Return $Similarity_{final}(c, c')$;
End.

V. RESULTS

The tests have been carried out with the data of the Ontology Alignment Evaluation Initiative 2007, benchmark where the selected tests ontologies are taken from the benchmark base put at the disposal of the scientific community by EON [11]. Our experiments are restricted to the following metrics that evaluate the goodness of the algorithm output and which are derivatives of well-known metrics from the information retrieval domain [12]: Precision, Recall, and FMeasure. The mapping algorithm has been implemented in java as a Protege PlugIn.

A. Tests 101-104

XMap++ has no problem handling the language generalization (test 103) and language restriction (test 104) features in the tests. The results (see result Table 1) show that the developed mapping algorithm enables us to achieve 0.98% precision and 100% recall in tests 101, 103 and 104. The test 102 also shows the performance of the algorithm, where, no alignments were retrieved by the system. Having no outcome it is not possible to calculate the presented evaluation measures. This is the expected outcome when trying to align two ontologies of completely different domains. Table 1 resumes the obtained results for tests 101 to 104.

TABLE I. ALIGNMENT RESULT FOR TESTS 101 – 104.

Test	**Name**	**Prec.**	**Rec.**
101	Reference alignment	0.98	1.00
102	Irrelevant ontology	NaN	NaN
103	Language generalization	0.98	1.00
104	Language restriction	0.98	1.00
	Average	0.98	1.00

B. Tests 201-204

Alignment tests 201 to 204 manipulate names and comments. The ontology 201 does not contain names and the ontology 202 contains neither names nor comments, so we will not consider the results of these tests. In fact, XMap++ considers concept and property IDs (identified by the "*rdf:ID*" tag) as well as their labels (extracted from "*rdfs:label*" tag), therefore the only information that can be used to create these result mappings in test 201 are comments, however the developed algorithm does not them (See result Table 2).

TABLE II. ALIGNMENT RESULT FOR TESTS 201 – 204

Test	**Name**	**Prec**	**Rec**
201	No names	0.11	0.01
202	No names, no comments	0.11	0.11
203	No comments	0.96	1.00
204	Naming conventions	0.94	0.86
	Average	0.53	0.49

Concerning tests 203 and 204, our mapping algorithm creates the mapping with high precision (see result Table 2). Recall values are also considerable. Despite the use of different naming conventions in the ontology 204, XMap++ is able to found all matches.

C. *Tests 205-210*

Both ontologies 205 and 209 were mapped with good precision but the recall scale remains relatively low (see result Table 3).

Concerning the test case 205, the weakness of the recall by the fact is explained through the searching for WordNet synonyms based on full labels. The percentages of precision and recall of the second test case are a bit lower than the ones in the first test case. This note goes to show that the matchers, which deal with labels, have a part in the success of mapping. For 205 and 209 we had expected that using WordNet would be an advantage.

TABLE III. ALIGNMENT RESULT FOR TESTS 205 – 210

Test	Name	Prec	Rec
205	Synonyms	0.57	0.12
206	Translation (name)	0.67	0.23
207	Translation (name and comments)	0.67	0.23
208	Naming conventions, no comments	0.94	0.86
209	Synonyms, no comments	0.57	0.12
210	Translation, no comments	0.67	0.23
	Average	0.68	0.29

The algorithm generates quite good mappings for ontologies 206, 207, and 210 with extreme precision and quite satisfactory recall. The results depicted in Table 3 show that the precision and recall are the same for the three tests which can be explained by the following. On the one hand, the fact of keeping or suppressing comments does not have an effect on the produced mappings as the algorithm doesn't make use of this information. On the other hand, since the labels are translated to French, so the matchers, which deal with labels, are faced with a situation of total ignorance. We conclude that the difference in language between ontologies affects the mapping. Ontology 205 which contains synonyms were mapped with high precision but with really weak recall what can be explained by the fact that our algorithm looks for WordNet synonyms based on the full terms from the ontologies so e.g. MastersThesis or MScThesis as one word does not have WordNet synonym but MSc and Thesis separately do.

The test case 208 is similar to the test case 204 where the name of each entity is replaced by another one with different conventions (see result Table 3).

D. *Tests 221-247*

These alignment tests manipulate hierarchy. The overall performance of XMap++ is good with any kind of hierarchy manipulation (no specialization, flattened hierarchy and expanded hierarchy). However, XMap++ alignment results for tests 228, 233, 236, 239, 240, 241, 246 and 247 are poor when the properties are suppressed from the tests as displayed in Table 4. This result confirms that our algorithm takes both syntactic and semantic similarity into account.

TABLE IV. ALIGNMENT RESULT FOR TESTS 221 – 247

Test	Name	Prec	Rec
221	No specialisation	0.89	0.72
222	Flatenned hierachy	0.96	1
223	Expanded hierarchy	0.96	1
224	No instance	0.95	1
225	No restrictions	0.93	1
228	No properties	0.46	0.19
230	Flatenned classes	0.92	1
231	Expanded classes	0.96	1
232	No specialisation, no instance	0.97	0.82
233	No specialisation, no properties	0.46	0.19
236	No instance, no properties	0.46	0.19
237	Flatenned hierachy, no instance	0.95	1
238	Expanded hierachy, no instance	0.95	1
239	Flatenned hierachy, no properties	0.52	0.34
240	Expanded hierachy, no properties	0.52	0.30
241	No specialisation, no instance, no properties	0.53	0.29
246	Flatenned hierachy, no instance, no properties	0.53	0.31
247	Expanded hierachy, no instance, no properties	0.53	0.29
	Average	0.74	0.64

E. *Tests 248-266*

The alignment tests manipulate hierarchy, labels and comments. The results have proven that the XMAP++ strictly relies on ontological concepts and properties name for mapping ontologies, therefore these tests did not produce any sensible mapping result.

F. *Tests 248-266*

The result mappings produced by the algorithm for this test battery is of high precision (see Table 5). The recall is high for the test 302 and relatively good for the test 304, but ontology 301 was mapped with weak recall. More in detail, the weakness in the recall of the test 301 is in the mapping of datatype properties. This is due to some reasons that affect the execution of some matchers, such as the difference in the hierarchies between the ontologies in the test 301 and its labels generally use the term "has", i.e. "hasNAME" instead of "NAME" (our approach does not split the strings into individual terms). Concerning the ontology 303, it was not mapped at all by the algorithm.

TABLE V. ALIGNMENT RESULT FOR TESTS 301 – 304

Test	Name	Prec	Rec
301	BibTeX/MIT	0.88	0.39
302	BibTeX/UMBC	0.9	0.4
304	INRIA	0.92	0.88
	Average	0.90	0.69

We have chosen the alignments generated by the four best matchers that have participated in the 2010 OAEI conference track [8]. The results are goods (see result Table 6), where it can also be seeing, that other algorithms give a better average than the one given by XMAP++, which is normal, since these algorithms find especially all the matches found.

TABLE VI. COMPARISON WITH OTHER TOOLS

system	AgrMaker		ASMOV		Ef2Match		XMap++	
test	**Prec**	**Rec**	**Prec**	**Rec**	**Prec**	**Rec**	**Prec**	**Rec**
1xx	0.98	1.00	1.00	1.00	1.00	1.00	0.98	1.00
2xx	0.95	0.84	0.99	0.89	0.98	0.63	0.65	0.47
3xx	0.88	0.58	0.88	0.84	0.92	0.75	0.90	0.69

VI. CONCLUSIONS

Overall, our preliminary experiments show that a combined method may underperform a single strategy in some cases. We have proposed a new framework called XMap++, which automatically determines, which ontology alignment methods to be used, what kinds of information to use in the similarity calculation and how to combine multiple methods as necessary.

The results obtained with our XMap++ tool turned out to be good, especially including dynamic multi-strategy alignment available in this version of the system which is still subject to improvements. The used benchmark helped greatly identify the power and weaknesses of the algorithm. In our future work, we will tend to investigate different horizons along with the enhancement of XMap++ by different additive elements to take into account full labels with no synonyms and language differences.

REFERENCES

[1] Z. Bellahsene and F. Duchateau, “Tuning for schema matching,” In: Bellahsene, Z, Bonifati A, Rahm E (eds): Schema Matching and Mapping. Springer Data-Centric Systems and Applications Series, 2011.

[2] M. Ehrig, S. Staab and Y. Sure, “Bootstrapping ontology alignment methods with APFEL,” Proc. Int. Conf. Semantic Web (ICSW), Springer LNCS 3729, 2005.

[3] Y. Lee, Sayyadian, M. Doan and A. Rosenthal, “eTuner: tuning schema matching software using synthetic scenarios,” VLDB Journal 16 (1): pp. 97–122, 2007.

[4] F. Duchateau, R. Coletta, Z. Bellahsene and R.J. Miller, “(Not) yet another matcher,” Proc. CIKM, poster paper , 2009.

[5] W. Djeddi and M.T. Khadir, “XMAP : A novel structural approach for alignment of OWL-Full ontologies,” International Conference on Machine and Web Intelligence (ICMWI), October 3, Algeria (2010), pp. 347–352.

[6] T. Gruber, “A Translation approach to portable ontologies,” Knowledge Acquisition, Vol. 5, no. 2, pp. 199–220, 1993.

[7] A. Maedche, “Ontology learning for the sematic web,” Kluwer Academic Publishers, 2002.

[8] Jéôme Euzenat and al., “Firsts results of the ontology alignment evaluation initiative 2010,” Ontology Matching OM-2010, pp. 85–120, 2010.

[9] T.L. Bach, R. Dieng-Kuntz and F. Gandon, “On ontology matching problems - for building a corporate semantic web in a multi-communities organization,” ICEIS (4): pp. 236–243, 2004.

[10] R. Ferrer-i-Cancho, “The structure of syntactic dependency networks: insights from recent advances in network theory,” In Levickij V. and Altmman G., editors, Problems of quantitative linguistics, pp. 60–75, 2005.

[11] Eon, “EON 2007: Evaluation of Ontology for the Web,” In 5th International EON Workshop, Ontology Alignment Evaluation Initiative Test library, 2007.

[12] H. Do, S. Melnika and E. Rahm, “Comparison of schema matching evaluations,” In Web, Web-Services, and Database Systems, Springer LNCS 2593, pp. 221–237, 2002.