A Novel Approach Using Context-Based Measure for Matching Large Scale Ontologies

Warith Eddine Djeddi and Mohamed Tarek Khadir

LabGED, Computer Science Department, University Badji Mokhtar,
Po-Box 12, 23000 Annaba, Algeria
{djeddi,khadir}@labged.net

Abstract. Identifying alignments between ontologies has become a central knowledge engineering activity. In ontology matching the same word placed in different textual contexts assumes completely different meanings. This paper proposes an algorithm for ontologies alignment named XMap ++ (eXtensible Mapping), applied in an ontology mapping context. In XMap++ the measurement of lexical similarity in ontology matching is performed using synset, defined in WordNet. In our approach, the similarity between two entities of different ontologies is evaluated not only by investigating the semantics of the entities names, but also taking into account the context, through which the effective meaning is described. We provide experimental results that measure the accuracy of our algorithm based on our participation with two versions (XMapSig and XMapGen) at the OAEI campaign 2013.

Keywords: Ontology alignment, WordNet's synset, context measures, semantic relationships.

1 Introduction

Ontology matching is one of the most plausible solutions to cope with heterogeneity problems in ontological contents [1]. Ontology matching refers to the process of finding relations or correspondences between similar elements of different ontologies.

The problem of finding the semantic mappings between two given ontologies lies at the heart of numerous information processing applications. Virtually any application that involves multiple ontologies must establish semantic mappings among them using external resources like domain ontology, corpus, thesaurus (e.g., WordNet, Wikipedia), to ensure interoperability. Algorithms that are used in ontology alignment may be very complex and may contain many features and parameters that can affect the performance even of commonly accepted and string metrics, when they are used in new contexts.

The context is the set of information (partly) characterizing the situation of some entity [2]. The notion of context is not universal but relative to some situation, task or application [3][4]. Easy access to this information is essential in enabling the user to verify candidate mappings. In particular, the neighborhood of a term (immediate parent and children in the "is-a" hierarchy) may be especially important. Understanding the

L. Bellatreche and M.K. Mohania (Eds.): DaWaK 2014, LNCS 8646, pp. 320–331, 2014.

correct meaning of words, that have different meanings in different contexts, reduces myriad of semantic problem (i.e., the polysemy and synonymy problem).

In this paper, in order to deal with lexical ambiguity, we introduce the notion of scope belonging to a concept which represents the context where it is placed. WordNet [5] is the semantic networks (thesaurus) exploited in our approach. The similarity between two entities of different ontologies is evaluated not only by investigating the semantics of the entities names, but also taking into account the local context, through which the effective meaning is described. Increasing the radius means enlarging the scope (i.e. this area) and, consequently, the set of neighbour concepts that intervene in the description of the context. Next, we present our two flexible and self-configuring matching tools XMapGen (eXtensible Mapping using Genetic) and XMapSig (eXtensible Mapping using Sigmoid) [6]. Then, we provide the results of our experiments on the OWL ontologies of OAEI campaign 2013 and we compare these results with those of other algorithms that competed in the OAEI challenge.

The rest of this paper is organized as follows. Section 2 reviews some related work in the area of ontology alignment. Section 3 presents in details, the novel approach using context-based measure. Section 4 defines the matching process strategy. Section 5 describes our evaluation methodology and discusses experimental results. Finally, section 6 concludes with an outline of future work.

2 Related Work

Many similarity measures have been adapted for use in matchers in various categories depending on the context of the similarity measurement, such as lexical, structural, or extensional matchers [7]. Lexical database such as WordNet [5] have been used to find synonyms for differing concept string labels. The string-based matchers then work not only on the specific concept labels but also on the corresponding synonyms in the WordNet Lexicon [8].

In [9], the authors introduce two techniques, which exploit the ontological context (the analysis of contexts based on the recurrence of nearby components) of the matched and anchor terms, and the information provided by WordNet, can be used to filter out mappings resulting from the incorrect anchoring of ambiguous terms.

BOwL exploits Lesk algorithm [10] for tagging each word belonging to an entity name with its most likely sense. During the matching stage, semantic and Boolean tags are exploited for obtaining effective mappings: reliable semantic tags are used during the ontology matching stage for identifying homonyms which do not share the same meaning, whereas Boolean tags are exploited for matching composite entity names as if they were Boolean propositions.

S-Match [11] works on lightweight ontologies, namely graph structures where each node label is translated into propositional description logic (DL) formula, which univocally codifies the meaning of the node, and then a propositional satisfiability (SAT) solver is used to check the validity of these formulas.The output of S-Match is a set of semantic correspondences called mappings attached with one of the following semantic relations: disjointness ($\perp$), equivalence ($\equiv$), more specific ($\subseteq$) and less specific ($\supseteq$). S-Match is extendable to host new algorithms and uses a predefined set of background knowledge sources, such as WordNet and UMLS.

More recent systems incorporate background knowledge sources to improve the ontology alignment process [12].

Regarding the proposed techniques devoted to develop highly sophisticated tools for performing ontology matching, we propose an ontology matching system that uses a technique for resolving the ambiguity of concepts names by taking the context of a concept into account. The context of a concept is defined by the set of those concepts that are close to the concepts in terms of shortest path. The large size of existing ontologies and the application of complex match strategies for obtaining high quality mappings makes ontology matching a resource- and time-intensive process, so most of the proposed ontology matching system rely on merely a defined depth (e.g, depth = 2). However the more interesting focus in our technique is to explore more in depth all the composed concepts that are connected directly or indirectly to the central node while taking into account the size of the compared ontologies. Finally to overcome the problem of computationally when aligning medium-sized and large-scale ontologies, we use a particular parallel matching on multiple cores or machines for dealing with the scalability issue.

3 Ontology Matching

Ontology matching tries to establish semantic relations between similar elements in different ontologies to provide interoperability. Ontology matching takes a pair of ontologies as an input and creates the semantic correspondence relationships between these ontologies.

3.1 Exploring Ontological Context of a Concept

This approach aims at discovering linguistic similarities between the involved entities. In general, linguistic similarities are based on morphology and semantics, which are associated to the words that describe the relative entities.

Often the same word placed in different textual contexts assumes completely different meanings. In addition, lexicons are not able to disambiguate situations in which homonyms occur. In order to deal with lexical ambiguity, this approach introduces the notion of "scope" of a concept which represents the context where the concept is placed.

Definition 1. *Let O be ontology and $c \in O$. The scope of c, with radius r, $scope(c,r)$ is a set of all the concepts outgoing from c included in a path of length r, with center c. More formally:*

$$scope(c,r) = \left\{ c^{'} | c^{'} \in O,\ dist(c,c^{'}) < r \right\}, \tag{1}$$

where $dist(c,c^{'})$ is the number of edges that are in the path from concept c to concept $c^{'}$. Let us note that $dist(c,c^{'}) = 0$, when $c = c^{'}$.

Definition 2. *Let α and β respectively concepts of the ontology O and $O^{'}$. Let $Name_{\alpha} = label(\alpha)$ and $Name_{\beta} = label(\beta)$ be respectively the linguistic labels associated to the concept α and β. Let $lex(Name_{\alpha}, Name_{\beta}) \in [0,1]$ be a lexical similarity associated to* *the pair of concept names α and β, with $\alpha \in O$, $\beta \in O^{'}$. The set L is composed of all pairs, defined as follows:*

$$\begin{aligned} L = \left\{ \left(\alpha^{'}, \beta^{'}\right) | \forall \alpha \in scope\left(\alpha^{'},\ r\right), \forall \beta \in scope\left(\beta^{'},\ r\right) \right\} \\ and \left\{ \exists lex\left(Name_{\alpha}, Name_{\beta}\right) \neq 0 \right\}. \end{aligned} \tag{2}$$

“The need to determine the degree of semantic relatedness between lexically expressed concepts is a problem that pervades much of the computational linguistics” [13]. Recent research on the topic in computational linguistics has emphasized the perspective of semantic relatedness of two words in a lexical resource, or its inverse semantic distance. A natural way to compute the similarity measure of words given a semantic network is to evaluate the distance of nodes corresponding to words being compared to the shortest path from one node to another, the more related the words are. Thus the length of the shortest path in a semantic network is named semantic distance. The first step towards defining a method for measuring the semantic similarity between a pair of concepts (or terms) using the lexicon WordNet is to define how the distance between two WordNet synsets can be measured. The following three similarity measures are based on path lengths between a pair of concepts: *lch* [14], *wup* [15], and *path*. *lch* finds the shortest path between two concepts, and scales that value by the maximum path length found in the is-a hierarchy in which they occur. *wup* finds the depth of the least common subsume (lcs) of the concepts, and then scales that by the sum of the depths of the individual concepts. The depth of a concept is simply its distance to the root node. The measure *path* is a baseline that is equal to the inverse of the shortest path between two concepts. We have adopted the Leacock-Chodorow (*lch*) and Wu and Palmer (*wup*) conceptual distance measures.

Leacock & Chodorow (*lch*) propose a normalized path-length measure which takes into account the depth of the taxonomy in which the concepts are found [14].

$$lch(c_1, c_2) = -log\frac{length(c_1, c_2)}{2D}, \tag{3}$$

where $length(c_1, c_2)$ is the number of nodes along the shortest path between the two nodes (as given by the edge counting method), and D is the maximum depth of the taxonomy.

Wu & Palmer (*wup*) present instead a scaled measure which takes into account the depth of the nodes together with the depth of their *least common subsumer* (*lcs*).

$$wup(c_1, c_2) = \frac{depth(lcs(c_1, c_2))}{depth(c_1) + depth(c_2)}. \tag{4}$$

Definition 3. *Let c_1 and c_2 be two ontology concepts in the ontology O. The distance between two nodes $\delta(c_1, c_2)$ is represented by the minimum number of edges that connect them.*

Definition 4. *Given an ontology O formed by a set of nodes and a root node R. Let c_1 and c_2 be two ontology concepts of which we will calculate the similarity. Then, g is* *the common ancestor of c and c'. The Wu-Palmer similarity is defined by the following expression [15]:*

$$sim(c_1, c_2) = \frac{2 \times \delta(g, R)}{\delta(c_1, R) + \delta(c_2, R)}. \quad (5)$$

3.2 Sense Disambiguation to Improve Matching

Through the given definitions, it is possible to individuate the meaning of a name associated to an ontology concept. Given a word, WordNet provides a list of all the synsets and word senses, related to that word. Firstly the following pseudo-code details the *Algorithm 1* which discriminate the actual sense of a word associated to a given concept.

```
input  : Ontology $O$, concept $\alpha \in O$, radius r and the word w
output: important synset of w which express the exact meaning
build an Array $T\left[|synset(w)|\right]$;
foreach $t_1$ in $synset(w)$ do
    Initialize $V\left[w,\ similarity(t_1)\right] = 0$;
    for $i \leftarrow 1$ to $r$ do
        Set $M = \oslash$;
        Set $GM = \oslash$;
        foreach $c$ in $scope(\alpha,\ i) - scope(\alpha,\ i-1)$ do
            // Collection of similarity values
            Set $S = \oslash$;
            foreach $t_2$ in $synset(label(c))$ do
                $sim \leftarrow similarity\&lch(t_1,\ t_2)$;
                $S \leftarrow S \cup (sim)$;
            end
            $max \leftarrow$ maximum in $S$;
            $max \leftarrow \frac{max}{[scope(\alpha,\ i)]}$;
            $M \leftarrow M \cup (max)$;
        end
        $GM \leftarrow$ maximum in $M$;
        $GM \leftarrow \frac{GM}{i}$;
        $V\left[w,\ similarity(t_1)\right] \leftarrow GM$;
    end
end
result $\leftarrow$ synset of the word $w$ with highest similarity in $V\left[\ \right]$;
return result;
```

Algorithm 1. Exploring the true sense of $\alpha \in O$ is determined by its place in the hierarchy of its ontology up to a certain depth.

The *Algorithm 1* takes as input an ontology O, a reference concept α in that ontology and a word w. The word w represents the name associated to the concept α (i.e. $w = label(\alpha)$). Secondly, the pseudo-code details also the *Algorithm 1* which takes as input

```
   input : Ontology $O$, concept $\alpha \in O$
           Ontology $O^{'}$, concept $\beta \in O^{'}$
   output: similarities number of $t_1$ compared to $t_2$

   // important synset of $\alpha$ discovered by Algorithm 1
1  $t_1 \in synset(\alpha)$;
   // important synset of $\beta$ discovered by Algorithm 1
2  $t_2 \in synset(\beta)$;
3  $sim \leftarrow similarity\&wup(t_1, t_2)$;
4  return sim;
```

Algorithm 2. Comparing the two synsets associated for a concept $c \in O$ and a concept $c' \in O'$

the output of the *Algorithm 1* (The two selected synset of concept c and $c^{'}$), and calculate the similarities between the two synset using the measure proposed by Wu and Palmer [15]. Let us note the semantic difference between the concept (or class) in an ontology and the label associated to that concept. The algorithms replies to question like *"I would like to know the effective sense of the word w, placed in the context (or scope) of the concept α"*.

First step (line 1) is to declare a vector structure whose size corresponds to the number of synsets (or senses) associated to the given word w. Goal is to maintain in each cell of the vector a pertinence value that represents how much the word w is semantically related to that sense (or belongs to that synset). *Algorithm 1* selects all the concepts in the scope of α (belonging to the reference ontology O) by varying the radius (lines 4-7), in order to get different set of terms. Then, the lch similarity between two terms coming from the concept name of α and the word w (line 10) is computed. For each concept c in the ring shaped area (computed as the difference of the areas between two successive radius, see line 7), the max similarity values between the synset name associated to c and a concept name in synset of α are maintained in M (line 15). At the end of the three loops (lines 5-16) the vector $V\left[w,\ similarity(t_1)\right]$ contains the max similarity values in the variable GM computed for each couple of terms coming from the fixed term $t_1 \in synset(w)$ and all the terms which occur in the scope of α, with respect to the current radius (line 18). This is repeated for each $t_1 \in synset(w)$ in the synset of w. At the end, there is a value (line 19) in each cell of the vector V associated to each term in the synset of w, which the algorithm uses it to judge the important synset with the highest similarity number reflecting the correct meaning of concept α.

Algorithm 2 takes as input the two selected synset; $t_1 \in synset(\alpha)$ fom the ontology O and $t_2 \in synset(\beta)$ fom the ontology O', discovered by *Algorithm 1*, and calculate the similarities between them using the measure proposed by Wu and Palmer [15]. The final semantic similarity value is added to the linguistic matcher or structure matcher in order to enhance the semantic ambiguity during the comparison process of entities names.

4 Proposed Algorithm

4.1 Ontology Matching Process of XMap++

XMap++ is a system for ontology alignment that performs semantic similarity computations among terms of two given OWL ontologies. XMap++ view match as an operator that takes two graph-like structures (e.g., classifications, RDF schemas) and produces a mapping between the nodes of these graphs that correspond semantically to each other. Our semantic matching approach include three matchers. The *String Matcher* based on linguistic matching compares the textual descriptions of the concepts associated with the nodes (labels, names) of each ontology. The *Linguistic matcher* jointly aims at identifying words in the input strings, relaying on WordNet [5] which provide additional information towards unveiling mappings in cases where features such as labels are missing or in cases where names are replaced by random strings. Finally the *Structural matcher* aligns nodes based on their adjacency relationships. The relationships (e.g., subClassOf and is-a) that are frequently used in the ontology serve, at one hand, as the foundation of the structural matching [16]. On the other hand, the structural rules are used to extract the ontological context of each node, up to a certain depth (radius). This process is enriched by applying a transitive inference mechanism, in order to add more semantic information that is not explicit in the asserted ontologies.

Alignment suggestions are then determined by combining and filtering the results generated by one or more matchers. Their values are combined using a differents aggregation strategies (Weighted sum, ANN, GA, MAX, etc.) [16]. The filtering consists of retaining the pairs of terms with a similarity value above a certain threshold as align-

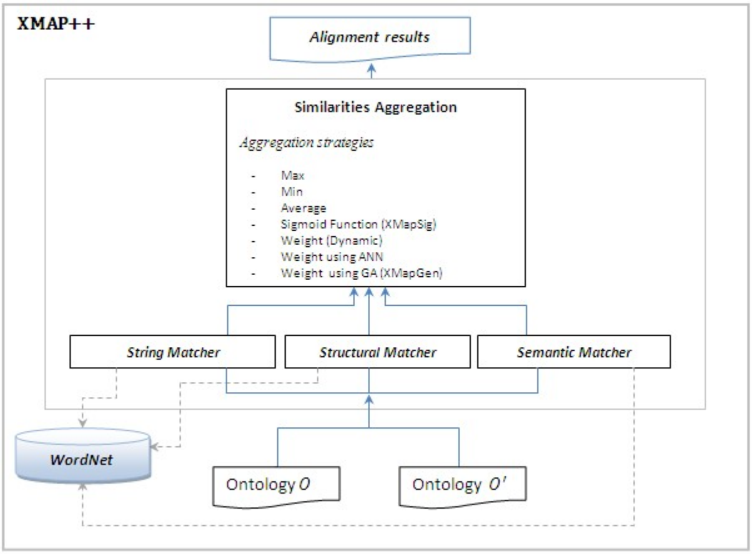

Fig. 1. Sketch of Architecture for XMap++

ment suggestions. The output of the alignment algorithm is a set of alignment relationships between terms from the source ontologies (see Fig. 1).

4.2 XMap++ Using WordNet's Synset

As expressed earlier in section 4.1, the ontology mapping outlines correspondences or matches between concepts coming from two different ontology. In this approach, the concept matching is measured by computing a similarity between concepts at linguistic level. The two algorithms described in the previous section has been exploited to evaluate the concept matching. More formally, given two ontologies O and $O^{'}$ and two concepts c and $c^{'}$, belonging respectively to these two ontologies, there is a match between c and $c^{'}$, if a similarity between them exists, computed as follows:

1. *Algorithm 1* is invocated twice: taking, in turn the concept c as an input and the ontology O, then by taking the concept $c^{'}$ and the ontology $O^{'}$. Outputs of these two independent executions are two indexes, i for the synset of c and $i^{'}$ for the synset of $c^{'}$. As said, they identify the important synset with highest similarity number that reflects the correct meaning of the two compared concepts.
2. Once discovered the two synset of involved concepts, the *Algorithm 2* compute the affinity between the concepts by a similarity measure between c and $c^{'}$ using the Wu-Palmer similarity.

In order to obtain all the semantic correspondences among the concepts in the two ontologies, this procedure can be applied for each pair of concepts coming from two ontologies. The final result is an ontology alignment; a similarity value is assigned to each discovered correspondence between concepts in the two ontologies.

5 Evaluation

5.1 Implementation and Setting

A detailed description of the implementaion of XMapGen and XMapSig is out of the scope of this paper, but we summarize here their principles characteristics : XMapGen and XMapSig, a new and lighter implementations of their ancestor XMap++ [16]. XMapGen uses Genetic Algorithm (GA) as a machine learning-based method to ascertain how to combine multiple similarity measures into a single aggregated metric. However XMapSig uses sigmoid function [16] for combining the corresponding weights for different semantic aspects, reflecting their different importance.

5.2 Data Sets and Evaluation Criteria

To evaluate our approach in OAEI 2013[1], we participated in five tracks including *Benchmark*, *Conference*, *Library*, *Anatomy* and *Large Biomedical Ontologies* tracks (see Table .1). We follow the evaluation criteria of OAEI, calculating the precision, recall and f-measure of each test case.

Table 1. Statistics of the OAEI 2013 tracks

Test set	Comparison size	depth (max)
Benchmark (Biblio)	$\cong$ 9409 element pairs	8
Anatomy Track	$(2.800\times3.300) \cong 9\cdot10^6$element pairs	17
Conference Track	$\cong$ 11025 element pairs	7
Library Track	$(25{,}000\times12{,}000) \cong 300\cdot10^6$element pairs	64
FMA-NCI small fragments	$(3{,}696\times6{,}488) \cong 10\cdot10^6$element pairs	20
FMA-SNOMED small fragments	$(10{,}157\times13{,}412) \cong 310\cdot10^6$element pairs	20

Table 2. Depth value variation and its effect on f-measure of XMapSig and XMapGen

	biblio	Anatomy	Conference	Library	FMA/NCI	FMA/SNOMED
Baseline	0.41 (edna)	0.77 (StrEqv)	0.52 (StrEqv)	0.57 (MaPrefEN)	0.81 (Average)	0.55 (Average)
YAM++	0.89	0.90	0.71	0.74	0.91	0.84
MaasMatch	0.69	0.41	0.36	-	0.46	-
Depth = (all the depth)						
XmapSig	0.58	0.75	0.53	0.45	0.60	0.23
XmapGen	0.54	0.75	0.53	0.06	0.61	0.24
Depth = 6						
XmapSig	0.52	0.533	0.37	0.369	0.489	0.154
XmapGen	0.49	0.567	0.38	0.375	0.501	0.173
Depth = 2						
XmapSig	0.46	0.49	0.23	0.24	0.41	0.18
XmapGen	0.43	0.51	0.26	0.28	0.42	0.19
Depth = 0						
XmapSig	0.41	0.33	0.22	0.19	0.14	0.06
XmapGen	0.38	0.345	0.26	0.21	0.19	0.09

Table 3. Execution time (in seconds) of XMapSig and XMapGen vs. OAEI 2013 entrants

	biblio	Anatomy	Conference	Library	FMA/NCI	FMA/SNOMED
YAM++	702	62**	600	731.86	94	100
MaasMatch	173	8532	76	-	12,410	-
Depth = (all the depth)						
XMapSig	612	393	600	2914.167	1.477	11.720
XMapGen	594	403	600	3008.82	1.504	12.127

5.3 Experimental Results

To gauge the effect of our semantic verification process, we have run the experiments both using the full XMapGen and XMapSig implementation (comparing all the depth), as well as using a system varying the level of the depth in the ontological structure. To facilitate our analysis, we assign each concept within its hierarchy a depth value (e.g, depth =0, depth =2 and depth =6). The concepts at the top level of the hierarchy

[1] http://OAEI2013.ontologymatching.org/2013/

(depth=0) do not possess any parent concepts. Then, the depth of the concept *C* is determined as the length of the path from *C* to the top level concept that is associated with *C* using the "subClassOf" relation.

The results of these experiments are shown in Table .2. It can be clearly seen that, as expected, the accuracy of our two systems decreases as the value of the depth decreases. It can clearly be seen that the use of all the depth in the hierarchy of a given ontology produces a more accurate result than the use of any less depth value. Whereas for a benchmark track the variation of the depth value has no effect on the f-measure, this may be a matter of just a few classes before the top level class is reached. Additionally, to tackle the large ontology matching problem we improved the runtime of the algorithm using a divide-and-conquer approach that can partition the execution of the matchers into small threads was improved and joins their results after each similarity calculation. A direct comparison between the XMapGen and XMapSig shows that the addition of GA does not has a negative effect on the algorithm but, on the contrary, leads to slightly better results, especially in terms of recall. In most tracks, XMapSig supplies high precision than XMapGen. Whereas using Genetic Algorithm (XMapGen) performs quite high in terms of recall than using sigmoid function (XMapSig)[6][17].

XMapGen and XMapSig produced fairly consistent alignments when matching the five tracks. Some reasons are related to: (a) the absence of domain and range definitions (in fact, of properties in general), as for anatomy, and the presence of multi-lingual labels; (b) fixed threshold used as a filter in the selection module. Different tests require different thresholds; (c) XMapSig does not respect languages, this may lead to false positives; (d) XMapGen and XMapSig exploit only the superclass-subclass relationships (subsumption relationships) that are frequently used in ontologies when the total number of entities is bigger than 1500 entities in each ontology. *We restrict the contextual similarity computing; only the value of the semantic relation between two concepts without taking in consideration the types of cardinality constraints and values between their properties.*

5.4 The Comparison against Other Systems in OAEI Campaign 2013

A detailed comparison against the 23 systems participated in the campaign OAEI 2013 is beyond the scope of this paper, but we selected two systems, YAM++ and MaasMatch, because they use WordNet as background knowledge [17].

We have tabulated the f-measure values for our two systems against YAM++ and MaasMatch in Table .2. As can be seen, the f-measure confirms the good performances of YAM++. In terms of baseline comparison, our two systems perform better than MaasMatch on 5 tracks. Whereas our two systems failed to provide an f-measure higher or roughly equal than the baseline for *Library* and *Large Biomedical* tracks. While MaasMatch performed worse than baseline for the *Anatomy*, *Conference*, *Library* and *Large Biomedical* tracks.

The results of the mappings runtime are presented in Table .3: a) *Benchmark track*, it ranges from less than a 3 minutes for MaasMatch and to nearly 11 minutes for both YAM++ and our two systems; b) *Anatomy track*, due to some software and hardware incompatibilities, YAM++ had to be run on a different machine and therefore its runtime (indicated by **) is not fully comparable to the other matchers. Meanwhile, the total time elapsed for the execution of our two systems was approximately 7 minutes against 2 hours and 22 minutes to MassMatch for finishing this track; c) *Conference track*, MaasMatch finished all 21 tests around 1 minute. Whereas, 10 minutes are enough for the three matchers (YAM++, XMapGen and XMapSiG); d) *Library track*, our two systems and YAM++ were able to generate an alignment within 12 hours. MaasMatch doesn't finish in the time frame; e) *Large biomedical ontologies (largebio)*, YAM++ provided the best results on FMA/NCI track 1 and FMA/SNOMED track 1.

Regarding performance we can conclude that YAM++ has a significant improvement from our two systems and MassMatch in terms of both f-measure and runtime, especially for very large scale tasks. This proves the effectiveness and efficiency of the strategy implemented by YAM++ which consists of implementing a disk-based method for storing the temporary information of the input ontology during the indexing process in order to save main memory space. Nevertheless, there is room for continued improvements in our algorithms because our approach lead to an important increase in precision, without having too negative an impact on recall which reflect the main objective of this approach [17].

Generally, according to our results in OAEI 2013, our two systems delivered fair results comparatively to other participants. The aim of this development experience was not to deliver a tool to compete with others in terms of precision and recall. Instead, we aimed at the development of a new and stable version of XMap++ using new and state-of-the-art technologies and alignment methods [17]. In case of scalability (see results of running tests on six dataset in [17]), our two systems are placed above eight out of twenty-three matchers participated in OAEI 2013. Matching larger ontologies still takes significantly longer time when parsing ontologies with Alignment API. We plan to solve this problem using an ontology parser which permits to load multiple ontologies in parallel via threading.

6 Conclusion and Future Works

This paper has presented a novel approach using context-based measure for semantic matching. The proposed ontology matching method based on the context of ontologies solves difficult problems of lexicography, polysemy and synonymy which occur in the phase of ontology mapping. The proposed method based on the variation of the radius value, which induces the variation of the scope area, in order to detect the correct meaning of the context description. The preliminary results were quite good to encourage us to continue seeking better solutions. In future work, how to customize the value of radius for each entity will be studied. That is, this research will be extended to build an algorithm that takes into account the type and the depth of taxonomies related to each entity, in order to automatically tune the radius value parameter. Moreover, our framework is very flexible: many semantic measures may be used in the future (JCN (Jiang and Conrath), Lin, Resnik, Lesk, hst(HIRST and ST-ONGE), etc.).

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
2. Dey, A., Salber, D., Abowd, G.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction 16, 97–166 (2001)
3. Dourish, P.: Seeking a foundation for context-aware computing. Human-Computer Interaction 16, 2–3 (2001)
4. Chalmers, M.: A Historical View of Context. Computer Supported Cooperative Work 13(3), 223–247 (2004)
5. Miller, G.: WordNet: An electronic Lexical Database. MIT Press (1998)
6. Djeddi, W., Khadir, M.T.: XMapGen and XMapSiG results for OAEI 2013. In: Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), pp. 203–210. CEUR-WS.org, Sydney (2013)
7. Sabou, M., Aquin, M., Motta, E.: Exploring the Semantic Web as Background Knowledge for Ontology Matching. J. Data Semantics 11, 156–190 (2008)
8. Lin, F., Sandkuhl, K.: A Survey of Exploiting WordNet in Ontology Matching. In: Bramer, M. (ed.) IFIP AI, pp. 341–350. Springer, Heidelberg (2008)
9. Gracia, J., Lopez, V., d'Aquin, M., Sabou, M., Motta, E., Mena, E.: Solving semantic ambiguity to improve semantic web based ontology matching. In: The 2nd International Workshop on Ontology Matching 2007, Busan, South Korea (November 11, 2007)
10. Mascardi, V., Locoro, A.: BOwL: exploiting Boolean operators and lesk algorithm for linking ontologies. In: Ossowski, S., Lecca, P. (eds.) SAC, pp. 398–400. ACM (2012)
11. Giunchiglia, F., Yatskevich, M., Shvaiko, P.: Semantic Matching: Algorithms and Implementation. J. Data Semantics 9, 1–38 (2007)
12. Zablith, F., d'Aquin, M., Sabou, M., Motta, E.: Investigating the use of background knowledge for assessing the relevance of statements to an ontology in ontology evolution. In: 3rd International Workshop on Ontology Dynamics (IWOD 2009) at ISWC-2009, Washington, DC, USA (2009)
13. Budanitsky, A., Hirst, G.: Semantic Distance in WordNet: An Experimental, Application oriented Evaluation of Five Measures. In: Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2000), Pittsburgh, PA (2001)
14. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 265–283. MIT Press (1998)
15. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, pp. 133–138 (1994)
16. Djeddi, W., Khadir, M.T.: Ontology alignment using artificial neural network for large-scale ontologies. the International Journal of Metadata, Semantics and Ontologies (IJMSO) 8(1), 75–92 (2013)
17. Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E.: Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), CEUR-WS.org, Sydney (2013)