DOI: http://dx.doi.org/10.26483/ijarcs.v11i4.6647

Volume 11, No. 4, July-August 2020

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

CROSS LINGUAL ONTOLOGY MATCHING BASED ON FUZZY SYNTACTIC MATCHING

Shubhi Shrivastava
Department of computer science
Banasthali Vidyapith
Rajasthan, India

Iti Mathur
Department of computer science
Banasthali Vidyapith
Rajasthan, India

Nisheeth Joshi
Department of computer science
Banasthali Vidyapith
Rajasthan, India

Abstract: Ontologies bolsters data disclosure, sharing and reuse among people and enable semantic interoperability between PC based structures. To develop correspondences between data thoughts addressed in Ontologies. Once in a while the correspondence between the client and PC is in various language, which is extremely hard to comprehend for both. Ontology matching is at the center of overseeing Cross Lingual on the semantic web. In this paper, we present a way to deal with take care of the issue of multilingualism on the semantic web, in view of Syntactic matching . To determine linguistic issue, two Ontologies (one in English and one in Hindi) of same space, interior portrayal, number of matching algorithm dependent on Syntactic method (Edit distance (Levenshteindistance LD)), and Machine Translator.

General Terms: Ontology Matching

Key Words: Cross Lingual Ontology Matching, Syntactic Matching, Edit Distance Algorithm, Fuzzy Implication, Machine Translation.

1. INTRODUCTION:

Ontologies have become key components in an assortment of information based applications. Be that as it may, they are constantly faced with the issue of heterogeneity { syntactic, phrased, theoretical or semantic. Ontology matching strategies propose answers for the heterogeneity issue via naturally finding correspondences between the components of two distinct Ontologies and in this way empowering interoperability [1,2].

In software engineering, estimated string matching (frequently informally alluded to as fuzzy string searching) is the method of discovering strings that coordinate an example roughly (as opposed to precisely). The issue of rough string coordinating is commonly separated into two sub-issues: finding surmised substring matches inside a given string and discovering word reference strings that coordinate the example around.

The closeness of a match is estimated as far as the quantity of crude activities important to change over the string into a careful match. This number is known as the Edit distancebetween the string and the patern. The typical primitive tasks are:

- insertion: cot → co**a**t
- deletion: co**a**t → cot
- substitution: co**a**t → co**s**t

These three tasks might be summed up as types of replacement by including a NULL character (here represented by *) any place a character has been erased or inserted:

- insertion: co*t → *co**a**t*
- deletion: *co**a**t* → co*t
- substitution: *co**a**t* → *co**s**t*

Some estimated matchers additionally treat transposition, in which the places of two letters in the string are traded, to be a primitive operation.

- transposition: *co**st*** → *co**ts***

Distinctive surmised matchers force various requirements. A few matchers utilize a solitary worldwide unweighted cost, that is, the complete number of primitive tasks important to change over the match to the example. For instance, if the example is coil, foil contrasts by one replacement, coils by one addition, oil by one cancellation, and foal by two replacements. On the off chance that all activities consider a solitary unit of cost and the breaking point is set to one, foil, coils, and oil will consider matches while foal won't.

We presentthe Cross Lingual Ontology matching based on Syntactic technique (Edit distance (Levenshtein distance LD)) and Bi-lingual Dictionary. Rest of the paper is organized as: Section 2 gives a brief description of the work done in the area of cross lingual ontology matching. Section 3 describes our approach; it explains the experimental setup and our methodology. Section 4 describes the evaluation procedure and Section 5 concludes the work done.

2. LITERATURE SURVEY:

Ontology matching, unlike other areas of computer science is still an unexplored territory. Though a lot of matchers have been proposed in the literature, only a handful of them have been pursued for further enhancements of both; the matching problem at large and the performance of the specific systems. In this section we shall provide a brief description of systems which have seen enhancements over time and would not be considering the ones which were used once upon a time but its current development is dormant.

Cruz at el. [6] have developed a matchers named Agreement Maker. This is considered as the best matchers as it has a very good user interface and a flexible architecture. This matchers involves the users of the system into matching process. Thus produces better results than any other matcher. The developers of this system believe that “users can help make better alignments which are not possible in automatic alignments.” Thus they prophesize the use of having semi-automatic matching systems. Ruiz and Grau [7] have developed LogMap at University of Oxford. This matcher incorporates logic based reasoning approach in their matcher. Since long ontologies have used description logic to reason out new concepts. Using it in matching process may be a very intuitive approach because it can produce better alignments. Though commenting on it is very early as the matcher is still in development stage and is yet to produce good results.

Jérôme [8] has developed a hybrid ontology matcher which can match the concepts and properties from two ontologies. He has used association rule paradigm [9] and statistical interestingness measure for implementing this matcher. Jorge et al. [10] have developed a matcher which tries to align ontologies using schema matching. For this, they applied two approaches, at first they extracted similar concepts and then applied different matching techniques onto the concepts extracted and finally produced aligned ontology. Peng et al. [11] have developed Lily, which may be a excellent matching system. It matches general and heavy-weight ontologies and produce good results for decent size ontologies, but it takes tons of your time to try to to so. At the core this matcher extracts semantic sub graphs and then tries to align it with other ontologies.

Juanzi [12] has developed RiMOM Ontology matcher which is one the highest performing matchers being tested in various evaluation campaigns, across the world . It is considered as a good matcher as it matches schema and instances available in the ontologies and uses multiple techniques to implement. Moreover to improve the results it also uses several external resources like WorldNet to do semantic matching. Fayçal et al. [13] have developed TaxoMap ontology matching system. This matcher can merge heavy-weight ontologies. It does so by finding correspondence between the concepts of two ontologies by applying subsumption, inverse and proximity relations. YAM++ [14] is another system which includes different matching algorithms which are combined to supply merged ontology. This system is self-configurable and extensible, as if the user is not satisfied with the results then he can provide his own customized matching approach. Mathur et al. [15] have developed a graph based ontology matcher which can use any one of the string matching algorithm to be combined with bi-partite graph matching.

CIDER-CL could also be a schema based ontology alignment system, which compares each pair of ontology entities on the thought of their similarity at different levels of their ontological context. This similarities are computed and combined through artificial neural networks. Both monolingual and cross lingual semantic analysis are used for comparison between different natural languages. Authors have presented some results of participation of CIDER-CL at OAEI’13 campaign. Thus proposed technique is suitable for monolingual matching with SoftTFIDF. And CL-ESA is suitable for cross lingual.[18] . LYAM++ may be a novel technique of aligning cross lingual ontology, which doesn't use MT method but uses the massive multilingual semantic network Babel net as a background . They also applies LYAM++ approach to new orchestration of the components of the matching workflow. Their proposed method believe automatic translation of labels to one target language or machine learning technique. They also demonstrated that his approach outperforms the simplest technique within the state-of-the-art[20]. XMAP is very scalable ontology matching system, which is automatically ready to adopt the matching task. it's highly scalable ontology matching system. It has used UMLs resources for discarding incorrect mappings and also implemented a cross- lingual ontology matching approach.[21]. In LYAM++ Let S and T be two input ontologies. Our goal is to align the previous (source) to the latter (target). Additionally, we assume that S is given during a tongue lS and T { during a language lT . we've chosen BabelNet as a source of background and our processing pipeline uses two matchers: a multilingual terminological matcher (the main matcher), making use of only two similarity measures, and a structural matcher.[22]. CroLOM may be a cross lingual ontology matching system. which describes about the system working how it applies NLP on each and each tongue . There after its translation phase using yandex translator, which uses English as a pivot language. FinallyCroLOM system computer the similarity between the translated entities.[23]. Agreement marker light(AML) is an automatic Ontology matching system. Its efficiency, extensibility, and skill to include external knowledge characterized it. It specialise in solving complex matching problems. it's supported finding the lexical similarities between source and target properties.[24].

OECM is a cross-lingual matching approach for ontology enrichment (OECM) so as to complement an ontology using another one during a di_erent tongue . A pro- totype for the proposed approach has been implemented and evaluated using the MultiFarm benchmark. Its supported terminological and structural matching. OECM outperforms all other systems in terms of precision, recall, and F-measure. For AML [24], authors include pre-computed dictionaries with translations, to beat the query limit of Microsoft Translator which decrease the e_ciency of their approach. LogMap [23] depends mainly on the initial mappings to get new mappings, which decreased after performing the interpretation . XMap[21] didn't achieve satisfactory results due to many internal exceptions. Surprisingly, we found seven new alignments, which didn't exist within the gold standard, when matching Conferencede with Ekawen, [25].

3. OUR APPROACH:

3.1. EXPERIMENTAL SETUP:

In order to experiment with the cross lingual Ontology matching, we need Ontologies in two different language. For this task we are considering English and Hindi language. So, we have developed both light weight and heavy weight Ontology on some different domain.

Some Ontologies are priory developed in some domain but its in only one language. So, we have developed some Ontologies: University Ontology, Tourism Ontology, Health Ontology, wine Ontology , Weather Ontology and many more. Some are light weight and some are heavy weight. All these Ontologies are in both the languages which we need for our experiment.

For matching Ontologies of same domain in two different language we have used Edit Distance algorithm to check the exact match. Also, some linguistic resources are used (WordNet). The Objective was to match the Ontology using linguistic resources.

3.2. METHODOLOGY:

As ontologies have a hierarchical structure where ideas, qualities and occurrences can be orchestrated in a tree like structure; using a graph matching algorithm here is far more intuitive mechanism. We have done Ontology matching using Bi-partite graph matching as suggested by mathur et al. [15].

Here, we have taken two Ontologies, one in English and one in Hindi. We have taken English Ontology as the source Ontology (Os) and Hindi Ontology as target Ontology as (Ot). The first step in our approach is to extract concepts, sub-concepts, properties of the Ontology source and target Ontologies. Next, we translated the extracted terms of source Ontologies into Hindi using shbadhkosh and a machine translation system developed by Joshi et al. [26][27].

Once this is done, the translated terms are matched with the terms extracted from the target Ontology.This is done using Edit Distance algorithm [28]. The algorithm searches for similarities between concepts, sub-concepts, properties and instances and are checked for equivalence, Isa correspondence and general correspondence. Thus all the matching is done using four tuples <x,y,r,t>. These are:

x ∈□ OS : x belongs to concepts, sub-concepts, properties and instances in source ontology.

y ∈□ Ot : y belongs to concepts, sub-concepts, properties and instances in target ontology.

r ∈□ R : r is a correspondence relations in a set of correspondence relations R, in our case these are Equivalence, Isa and General correpondence.

t ∈□ T: t is the similarity metric used in alignment from a set of available metrics T, in our case this are Levensthein Distance.

Using these mappings, we generated a score matrix in the following format:

$$s = \begin{bmatrix} O_{11}O_{21}O_{12}O_{21} \dots \dots . . O_{1n}O_{21} \\ O_{11}O_{22}O_{12}O_{22} \dots \dots . . O_{1n}O_{22} \\ \vdots \\ \vdots \\ \vdots \\ O_{11}O_{2m}O_{12}O_{2m} \dots \dots . . O_{1n}O_{2m} \end{bmatrix}$$

Here, $[O11O21]$ is the mapping between one of the elements (concepts, sub-concepts, properties, instances) of source ontology OS with one of the elements (concepts, sub-concepts, properties, instances) of target ontology Ot. This has the value which is produced by the similarity metric. For example, if we have two concepts as pen and लेखनी, then its score would be 1 and the similarity is calculated using the formula in equation 1.

$$sim(x, y) = \frac{\#edits(x, y)}{\max(len(x), len(y))}$$

Here x and y are the two strings, in our case x is "pen" and y is "लेखनी". #matches(x,y) is the no. of edits required to make the two strings equal and len(x) is the length of string x, len(y) is the length of string y. the maximum of the two is selected to compute the final score. This is done for all the mappings which then generate the score matrix of all the matched elements of both the ontologies. This matrix can be seen as bipartite graph which has two disjoint sets of vertices (in our case mapping elements of OS and Ot) and edge weights (similarity values) are clearly mentioned.

Once the score matrix is generated, it is passed to our graph matching algorithm. We used Hungarian method [18] for matching our score matrix (bipartite graph). This gave us the best matching pairs in the matrix which are then used to generate the aligned ontology. Figure 1 shows the architecture of our system. A snapshot of aligned ontology is shown in figure 2.

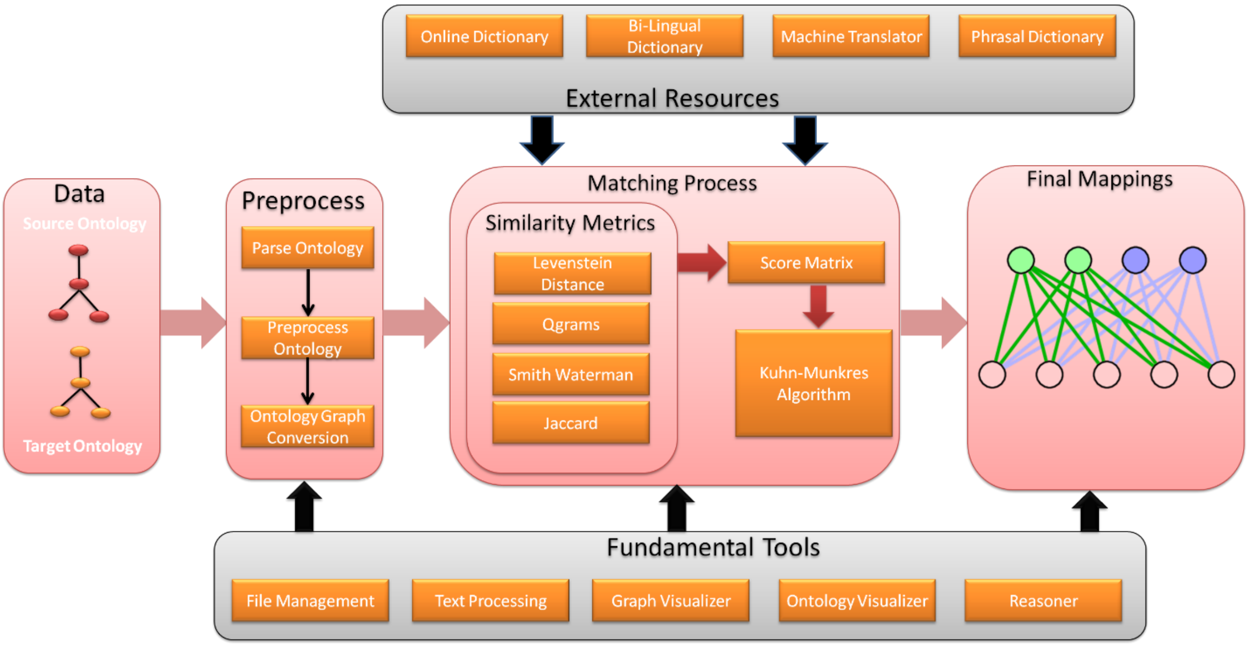
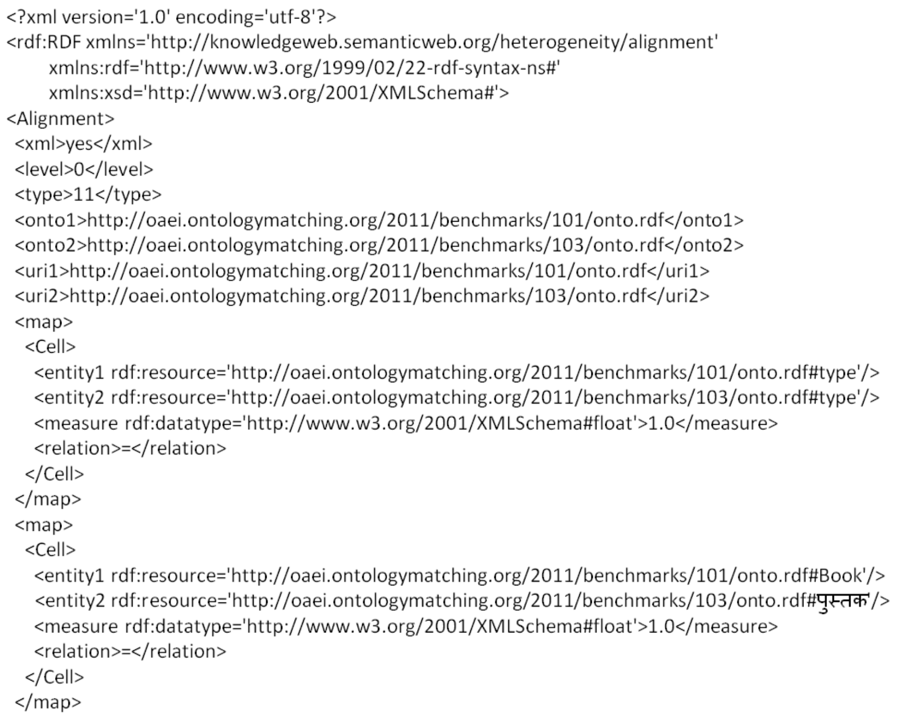


Figure 1: Architecture of Cross Lingual Ontology Matching System and Framework



```
<?xml version='1.0' encoding='utf-8'?>
<rdf:RDF xmlns='http://knowledgeweb.semanticweb.org/heterogeneity/alignment'
     xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
     xmlns:xsd='http://www.w3.org/2001/XMLSchema#'>
<Alignment>
 <xml>yes</xml>
 <level>0</level>
 <type>11</type>
 <onto1>http://oaei.ontologymatching.org/2011/benchmarks/101/onto.rdf</onto1>
 <onto2>http://oaei.ontologymatching.org/2011/benchmarks/103/onto.rdf</onto2>
 <uri1>http://oaei.ontologymatching.org/2011/benchmarks/101/onto.rdf</uri1>
 <uri2>http://oaei.ontologymatching.org/2011/benchmarks/103/onto.rdf</uri2>
 <map>
  <Cell>
   <entity1 rdf:resource='http://oaei.ontologymatching.org/2011/benchmarks/101/onto.rdf#type'/>
   <entity2 rdf:resource='http://oaei.ontologymatching.org/2011/benchmarks/103/onto.rdf#type'/>
   <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
   <relation>=</relation>
  </Cell>
 </map>
 <map>
  <Cell>
   <entity1 rdf:resource='http://oaei.ontologymatching.org/2011/benchmarks/101/onto.rdf#Book'/>
   <entity2 rdf:resource='http://oaei.ontologymatching.org/2011/benchmarks/103/onto.rdf#पुस्तक'/>
   <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
   <relation>=</relation>
  </Cell>
 </map>
```

Figure 2: A snapshot of aligned ontology

4. EVALUATION:

For checking the performance of Ontology Matchers, the two Ontologies shall be manually aligned and then it would compared using Precision and Recall and F-Measure, which are considered to be complete quality measures. Precision measures the ratio of found alignments that are correct. It relates to accuracy and is inverse to error rate. It is calculated using equation .

$$Precision\ (P) = \frac{Matc hes}{TotalMapsinMac\ \ hineAlignment} \quad (2)$$

Recall measures the ratio of correct alignment with the total correct existing alignments. It is calculated using equation 3.

$$Recall\ (R) = \frac{Matc\ hes}{TotalMapsinHumanAlignment}\ (3)$$

A high recall does not mean a complete quality alignment as it might also align incorrect alignments available in the manual match. On the other hand a high precision does not mean a high quality. In order to balance the two f-measure is used. It is calculated using equation 3.

$$F - Measure = \frac{2 \times P \times R}{P + R}\ (4)$$

We matched 10 Ontologies and found that in all cases the matcher produced high F-Score. Table 1 shows the results of this evaluation

Ontology Domain	Precision	Recall	F-Measure
Health	0.599843	0.598892	0.599367123
Tourism	0.381734	0.328588	0.353172819
University	0.417634	0.426212	0.421879401
School	0.598171	0.599667	0.598918066
Automobiles	0.187738	0.486169	0.270875345
Library	0.597077	0.599623	0.598347292
Automobiles	0.599495	0.599811	0.599652958
Finance	0.598039	0.599863	0.598949611
Furniture	0.599997	0.599979	0.599988
Electronics	0.079586	0.488888	0.136888021

5. CONCLUSION:

In this paper, we have shown the implementation of Fuzzy matching technique. We have used bipartite graph matching algorithm in creating aligned ontology. Next, we translated source Ontologies into Hindi using shbadhkosh and a machine translation system. Translated source Ontology is matched with the target Ontology using Edit Distance Algorithm. This approach found that the matcher produces high F-score.

6. REFERENCES:

[1] J. Euzenat and P. Shvaiko, Ontology matching. Heidelberg (DE): Springer-Verlag, 2007.

[2] D. Ngo and Z. Bellahsene, \YAM++ : A multi-strategy based approach for ontology matching task," in Knowledge Engineering and Knowledge Management -18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings, pp. 421{425, 2012.

[3] P. Shvaiko and J. Euzenat, \Ontology matching: state of the art and future challenges," Knowledge and Data Engineering, IEEE Transactions on, vol. 25, no. 1, pp. 158{176, 2013.

[4] D. Spohr, L. Hollink, and P. Cimiano, \A machine learning approach to multi-lingual and cross-lingual ontology matching," in The Semantic Web{ISWC 2011, pp. 665{680, Springer, 2011.

[5] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, \The agreementmakerlight ontology matching system," in On the Move to Meaningful Internet Systems: OTM 2013 Conferences, pp. 527{541, Springer, 2013.

[6] Cruz, I. F., Stroe, C., Caci, M., Caimi, F., Palmonari, M., Antonelli, F. P., Keles, U. C. 2010. Using AgreementMaker to Align Ontologies for OAEI 2010. Fifth International Workshop on Ontology Matching, co-located with the International Semantic Web Conference, Shanghai, China.

[7] Ruiz, E. J., & Grau, B. C. 2011. LogMap: Logic-based and Scalable Ontology Matching. In the 10th International Semantic Web Conference

[8] Jérôme, D. 2011. AROMA results for OAEI 2011. In Proceedings of the Sixth International Workshop on Ontology Matching.

[9] Agrawal, R., Imielinski, T., Swami, A. 1993. Mining association rules between sets of items in large databases. Vol 22 (2), ACM.

[10] Jorge, G., Bernad, J., Mena, E. 2011. Ontology matching with CIDER: evaluation report for OAEI 2011. In Proceedings of the Sixth International Workshop on Ontology Matching.

[11] Peng, W., Xu, B. 2008. Lily: Ontology alignment results for OAEI 2008. In Proceedings of the Third International Workshop on Ontology Matching.

[12] Juanzi, L., Tang, J., Li, Y., & Luo, Q. 2009. Rimom: A dynamic multistrategy ontology alignment framework. Knowledge and Data Engineering, IEEE Transactions on, Vol. 21(8), pp 1218-1232.

[13] Fayçal, H., Safar, B., Niraula, N.B., Reynaud, C. 2010. TaxoMap alignment and refinement modules: Results for OAEI 2010. In Proceedings of the Fifth International Workshop on Ontology Matching.

[14] DuyHoa, N., Bellahsene, Z. 2012. YAM++: a multi-strategy based approach for ontology matching task. Knowledge Engineering and Knowledge Management. Springer Berlin Heidelberg, pp 421-425.

[15] Mathur, I., Joshi, N., Darbari, H., Kumar, A. 2014. Shiva: A Framework for Graph Based Ontology Matching. International Journal of Computer Applications, Vol 87.

[16] Benjamins, V., Contreras, J., Corcho, O., Gómez-Pérez, A.: Six challenges for the semantic web. In: Special Interest Group onSemantic Web and Information Systems (SIGSEMIS Buelletin).(2004).

[17] MarouenKachroudi, Sami Zghal, and Sadok Ben Yahia, Using Linguistic Resource for Cross-Lingual Ontology Alignment. iJES – Volume 1, Issue 1, August 2013.

[18] Jorge Gracia and Kartik Asooja,"Monolingual and cross-lingual ontology matching with CIDER-CL: evaluation report for OAEI 2013", proceedings of the 8th international conference on ontology matching- volume 1111.

[19]Iti Mathur, Nisheeth Joshi, Hemant Darbari, Ajai Kumar "Shiva++: An Enhanced Graph based Ontology Matcher", International Journal of Computer Applications (0975 – 8887) Volume 92 – No.16, April 2014.

[20] Abdel Nasser Tigrine, Zohra Bellahsene, Konstantin todorov,"LYAM++ results for OAEI 2015-hal-Iirmm.ccsd.cnrs.Fr.

[21] WarithEddine DJEDDI, Mohamed Tarek Khadir and Sadok BENYAHIA, "XMap: results for OAEI 2015", OM 2015.

[22] Abdel Nasser Tigrine, Zohra Bellahsene, Konstantin Todorov," Light-weight Cross-lingual Ontology Matching with LYAM++",HAL Id: lirmm-01408025**,** Submitted on 2 Dec 2016.

[23] AbderrahmaneKhiat, "CroLOM results for OAEI 2017", OM 2017.

[24]Daniel Faria, Catia Pesquita, Booma SowkarthigaBalasubramani," Results of AML participation in OAEI 2018", OM 2018.

[25] Shimaa Ibrahim, Said Fathalla, Hamed ShariatYazdi, Jens Lehmann,andHajiraJabeen," OECM: A Cross-lingual Approach for Ontology Enrichment", 2019.

[26] Nisheeth Joshi, Hemant Darbari, and Iti Mathur,"Human and Automatic Evaluation of English to Hindi Machine Translation Systems", Advances in computer science, engineering and application, pp 423-432. 2012.

[27] Rashmi Gupta ,Nisheeth Joshi and Iti Mathur," ANALYSING QUALITY OF ENGLISH-HINDI MACHINE TRANSLATION ENGINE OUTPUTS USING BAYESIAN CLASSIFICATION", International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 4, No. 4, July 2013

[28] Levenshtein, V.I. 1966. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady. Vol. 10.