

Evaluation of String based Ontology Alignment Technique

Muhammad Sohail Nisar
University of Engineering &
Technology, Mardan, Pakistan

Muhammad Waqar
Bacha Khan University
Charsadda, KPK, Pakistan

Nouman Khan
University of Malakand
Chakdara, Pakistan.

ABSTRACT

To connect the semantic space between various related ontologies, computer scientists and specialists in the field of Semantic Web have suggested and executed different ontology matching systems. In any case, these strategies have a great space for improvement and continuous research should be done to get better results in terms of precision, recall and F-measure for the purpose of aligning ontologies. This work analysed various string similarity metrics for the task of ontology alignment whose definitions are taken from two libraries that are Alignment API and secondstring library. These metrics are used to align different ontologies provided with Alignment API package and then evaluated the performance of these metrics with regard to precision, recall and F-measure. Further, different ontology alignment systems are studied which used string similarity metrics in combination of structural similarity metrics and linguistic similarity metrics. The role of string similarity metrics is analysed and found that Jaccard in combination with TF/IDF similarity metric works good as well as ngram also works better.

Keywords

Ontology alignment, String Alignment, Evaluation, Precision and Recall

1. INTRODUCTION

Modern information systems require sharing and reusing of information between different applications and websites which led to the introduction of Semantic web. Semantic web goes for production of a typical system that enables information to be shared and reused crosswise over application, industry, and locale limits. Put another way, it is "a web of information connected up so as to be effectively processable by machines, on a worldwide scale". It can be described as to a productive approach to depict information all the more definitively for people and machines alike. Semantic web utilizes ontologies to characterize the ideas and connections used to portray a knowledge related to a specific domain. Ontology alignment can be described in various ways depending upon its purpose like it is the way toward bringing at least two ontologies into common understanding, making them compatible and lucid. Currently much research has been done regarding string based ontology alignment. Each of these techniques has its own merits and demerits but no such technique is available yet which gives semantically equivalent result.

2. RELATED WORK

Tim Berners-Lee, the founder of the web, initially imagined an entirely different world wide web than the existing today – which humans and computers could look for their required information [1]. Currently, researchers in a wide variety are putting their efforts to achieve this objective and ontology

alignment is one of the steps towards this objective. Describing concepts in a specific domain is done in an ontology and the way these concepts are interrelated [2]. It is common for a domain to be expressed by multiple ontologies, available on the internet, thus there is a possibility of using different terms for an entity in different ontologies or using same terms for different entities in distinct ontologies. To tackle this scenario, the two ontologies are aligned with each other to semantically correlate their corresponding entities. In ontology alignment, two ontologies are taken as input in which each comprises of classes and properties. The alignment process shows result in the form of relationship as equivalence / correspondences among entities from the two participating ontologies [1]. In a nutshell, ontology alignment can be said as a set of similarities among domain entities.

Entities belonging to different ontologies can be looked at through various ways. Because of this reason, diverse matchers can be useful in discovering similarity and correspondence among the entities from various ontologies. Similarities between entities contained in various ontologies can be quantified based on their structure, linguistic and string correspondences or relationship. The string methodology looks for arrangements of characters which is normally utilized for matching entities' names as well as labels and description of entities. Structural method measures similarities between classes, for example, super class, sub-class or properties of classes. The linguistic technique looks for the equivalent words and homonyms to discover closeness among elements in various ontologies [3-5].

There has been some earlier examination of string similarity measurements related to ontology alignment as a major aspect of the improvement of another string similarity technique composed exactly for this area done by Stoilos et al. [6]. They compared their particular metric to that of Smith-Waterman, Levenstein, Jaro Winkler, Needleman-Wunsch (a weighted adaptation of Levenstein), Monge Elkan, 3-gram, and substring using set of the OAEI benchmark tests. These sets benchmark tests was previously used by OAEI to track but was eliminated in 2010 for a dynamically created test set. Stoilos et al found Monge Elkan and Smith Waterman measurements ineffective on the test. The technique created by the specialists executed the best. Further, Information Web Consortium in 2004, in a report, described various string techniques in connection with ontology alignment [3]. String pre-processing methodologies like normalization and stemming are also examined in this report.

Exactly when the string similarity methodology is used for different domains, there are all the more interesting reviews. For instance, Branting examined string metrics as associated with the names of people, associations, and affiliations,

particularly in lawful cases [4]. Alignment of names can be classified in nine ways: accentuation, word changes, wrong spellings, abbreviations, word avoidances, upper casing, qualifiers, spacing and hierarchical terms. His work surveyed the execution of various blends of standardization, ordering (making sense of which names would be appeared differently in relation to each other) and similitude estimations. He saw that string standardization was useful for this application and that a string closeness metric that he called RWSA (delineated underneath) achieved the best outcomes. Besides, Cohen et al did an exceptionally detailed examination of string metrics that are associated with the assignment of matching names [5]. They found that on the data sets which they examined, the techniques of TF-IDF, Monge Elkan, and Soft TF-IDF performed best. Further, they built up the SecondString Java library to measure similarity in strings that has turned out to be broadly utilized by the researchers.

A couple of analysts have not intended to examine string similarity techniques but instead have come across some interesting results about the point while working on ontology alignment systems. For instance, the designers of Onto-Mapology utilized a number of string techniques like Jaro, Jaro-Winkler, TF-IDF, and Monge Elkan and found that Jaro-Winkler is having the most effective execution [7], and the engineers of SAMBO, used for biomedical ontologies, discovered that a weighted sum of these string metrics that are n-gram, alter separate, and a mysterious set metric worked better than any of those measurements alone [8]. Likewise, the X-SOM developers noticed that the ideal grouping of similarity metrics does not change in light of the domain of the ontologies rather than taking into account their design characteristics [9].

String similarity metrics is an older subject of research but which string similarity technique(s) works better for ontology alignment is still not clear. In the OAEI platform, algorithms used to examine 24 string similarity metrics. It was discovered that, Monge Elkan worked very well in matching names however it badly performed for ontology alignment, yet a few of the ontology alignment systems in the OAEI competition used Monge Elkan. Since almost all ontology alignment algorithms utilize a string similarity metric, more knowledge in it would be of benefit to numerous researchers. The work mentioned here extends the past efforts talked about above by considering a more extensive variety of string metrics, string preprocessing methodologies, and ontology types. It likewise takes the work further by setting the string metrics into a complete ontology alignment framework and comparing the outcome of that framework with the present state of the art.

3. STRING BASED ONTOLOGY ALIGNMENT TECHNIQUES

The only platform for doing research in ontology alignment was The OAEI (Ontology Alignment Evaluation Initiative). Since 2006, the OAEI competition requires its members to present a short research paper depicting their suggested methodology and results. These papers were thoroughly overviewed to check what type of metrics they had used and in addition what type of pre-processing techniques they had utilized (or suggested).

String similarity metrics can be grouped in three different ways:

- Perfect-sequence versus imperfect-sequence.
- set versus whole string
- global versus local

Global versus local tells how much information is needed by the metric to term a couple of strings to be a match or a non-match. A metric is global when, before matching of any string, the string metric process some data over the greater part of the strings in either one or both ontologies. Whereas, the local metric is, when the whole input is just the currently considered pair of strings. For the specific input pair of ontologies the global metrics may be better, but its time complexity will be increased.

In case of Perfect-sequence metrics, it will be declared as a match if characters are at the same position in both strings. On the other hand, in imperfect-sequence metrics, strings will match if they have matching characters and their locations in the strings must not differ by more than the threshold. Full length of the string is considered as threshold by some metrics. In situations where the word pattern of labels might vary, the imperfect-sequence metrics performs well. This is basic in scientific ontologies. For example, someone might want to match arm muscle with muscle of the arm.

Imperfect sequence measurements will most probably distinguish such matches. The disadvantage is that they likewise often bring about all the more false positives. For example, the imperfect-sequence metric will result in perfect match with the words stop and post if the limit is the whole length of the string. Further, there are set-based string matching techniques. These techniques find out how many the group of tokens in both strings overlap each other. Words inside the strings are considered as tokens. Only string method should be used by the set-based metric to check for equivalency of the tokens (or very close to be viewed as equivalent). For longer strings for example records or longer sentences, word-based set measurements does well whereas for shorter strings it yields high precision but low recall. Normally components in ontologies have shorter names which comprise of just one word or more, however certain domain ontologies may have extended labels. Additionally, in an ontology the names of people (as against to names of classes or properties) are normally of longer lengths. Word-based string matching techniques may do better.

Following is the list of string matching techniques found in the audit of OAEI members and are listed taking into account the groupings mentioned previously. One group does not have any methods from either non-set or global or perfect-sequence. These metrics were selected to show those mostly utilized as a part of existing alignment frameworks and additionally to thoroughly examine all mixes of the classification framework delivered.

- Set
 - Local
 - Imperfect-sequence
 - RWSA
 - Soft Jaccard (along with Levenstein)
 - Perfect-sequence
 - Overlap Coefficient (with exact)
 - Jaccard (with exact match)
 - Global
 - Perfect-sequence
 - Evidence Content (with exact)
 - TF-IDF (with exact match)
 - Imperfect-sequence

- Soft TF-IDF (with Jaro-Winkler)
- Non-set
 - Local
 - Perfect-sequence
 - Longest Substring Common
 - Exact Match
 - Substring test
 - Suffix
 - Prefix
 - Imperfect-sequence
 - Levenstein
 - Lin
 - Jaro
 - Stoilos
 - Monge Elkan
 - Smith Waterman
 - Smith Waterman Gotoh
 - Jaro-Winkler
 - N-gram
 - Normalized Hamming Distance
 - String Matching (SM)
 - Global
 - Perfect-sequence
 - None
 - Imperfect-sequence
 - COCLU

The essential thought behind each metric is described below.

3.1 Compression-based Clustering (COCLU)

COCLU uses a Huffman tree in order to group the strings of one ontology and after that matching every string in the second ontology to the proper cluster. Strings are viewed as equal which are part of the same cluster. The decision to place a new string in a certain cluster or make a new one is taken based on Cluster Code Difference (CCDi), a string distance metric which is calculated as for all strings in the cluster, adding the length of the Huffman codes and then adding the new string to the cluster and then adding again the length of the Huffman codes for all strings along with the new strings. Now the difference of the two values is Cluster Code Difference. This will result in collection of strings with the common characters, no matter what the sequence of those characters is. Details about COCLU can be read here in [10].

3.2 Exact Match

This is the clearest matching technique. For similar strings it basically returns one and otherwise zero.

3.3 Document Indexing

This methodology utilizes current record indexing as well as recovery mechanisms as a string matching technique. Every element of the second ontology to be aligned is dealt with as a record. The text of the record differs in various ways. Alternatives contain any blend of an entity's id, name, remark, title, predecessors, children, neighbors and examples. A typical web crawler like Lucene or Indri, initially, discovers the records (i.e. elements). The first matching ontology elements are now dealt with as search queries throughout the second matching ontology. Now the system forms the matches to the top indexed lists, given that the quality of the matches is over a threshold that the client sets.

3.4 Evidence Content

This metric and the Jaccard metric are the counterpart of each other. This metric does not weight every word. It weights each word taking into account their evidence content. Evidence content of each word can be found by checking that in how many elements of ontology the word exist, then finding the number of times that elements occur in ontology and finally taking negative logarithm of this number will yield evidence content. Its use in ontology alignment is described in [11].

3.5 Hamming Distance

It is measure of replacements necessary to convert one string into another. Distance is divided by the size of the string in the standard form of hamming distance. This metric differs from the Levenstein distance in only that it works on those strings which have the same length.

3.6 Jaccard

A typical string matching technique which has the following formula:

$$Jaccard(string1, string2) = \frac{|A \cap B|}{|A \cup B|}$$

This technique is often employed as a set metric in which the greater part of the distinctive words in the two matched strings is indicated by the union of words A and B whereas the intersection indicates those words which are in both strings (as found by basic string similarity metrics). This metric may also be used as a base metric instead of set metric by viewing distinct letters in the strings rather than words.

3.7 Jaro

An alternative basic string matching technique. It has the following equation:

$$Jaro(s1, s2) = 1/3 \left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \right)$$

Where quantity of matched characters is m and the measure of transpositions is t. Two strings will be considered as similar if these are not more different than the expression

$$\frac{\max(string1.length, string2.length)}{2} - 1$$

When two characters match but in reverse direction are said to be transpositions.

3.8 Jaro-Winkler

The idea behind this is that various comparative strings have the same roots but different endings. For example verbs and descriptive words. Its formula is:

$$JaroWinkler(s1, s2) = Jaro(s1, s2) + (lp(1 - Jaro(s1, s2)))$$

Length of the basic prefix which is normally equal to four characters is represented by l here, and weight of probability of similar prefix is represented by p (its value should be less than 0.25 and is generally set to 0.1).

3.9 Levenstein Edit Distance

This is certainly the most regularly utilized metric as a part of ontology alignment frameworks. It is the quantity of replacements, removals, and insertions necessary to change one string into another. Its value can be standardized by partitioning the edit distance with the string size (either the first string, to

make an asymmetrical methodology, or with average size of the two strings).

3.10 Lin

To find similarity between two distinct strings, the magnitude of similarity between these strings is divided by the degree of information it accepts to depict them. [12]

3.11 Monge Elkan

Both a set-based matching technique as well as a variation of the Smith-Waterman technique are explained by Monge and Elkan in their research work [13]. Various researchers term this as the Monge Elkan matching technique.

3.12 N-gram

Each string is changed into a group of n-grams. For example, for the word Pakistan with value of n is 3, the arrangement of n-grams would be {Pak, aki, kis, ist, sta, tan}. Any set-based matching technique then compares the resultant group of n-grams. An alternate way is to put special characters to specify the beginning and end of the string.

3.13 Overlap Coefficient

This metric is considered as the sibling of Jaccard metric. It has the formula

$$\text{overlap}(\text{string1}, \text{string2}) = \frac{|X \cap Y|}{\min(|X|, |Y|)}$$

Here X indicates group of tokens in one string whereas Y indicates in the other string.

3.14 RWSA

RWSA (Redundant, Word-by-word, Symmetrical, Approximate) relies on the system of classification for string matching measurements described in [4]. Soundex representation for each string's first and last word is used to order the strings. Soundex is a sound encoding containing the string initial character along with three digits illustrating the classifications based on sound of the following three consonants, in case they exist.

When matching two strings, hashing smaller of both strings will recover all possible matches, and the remaining algorithm keeps running on these possible matches to find the best correspondence and figure out whether it is over a threshold or not. The algorithm divides both strings into their segment words. If each word in the shorter string is similar to a word in the bigger string then the two strings are thought to be a match. If the edit distance is inside a mismatch limit then it will be an estimated match. To find the edit distance, the cost of replacement, deletion and insertion is equal to 1.0 and for transpositions it is 0.6. This metric as compared to others is utilized on bigger ontologies because of the property of indexing to recover potential correspondences.

3.15 Smith-Waterman

This metric was initially utilized to manipulate the distance between biological sequences. This metric is based on applying dynamic programming algorithm on matrix showing matches as well as distances between the matched strings.

3.16 SMOA (Stoilos Metric)

It was particularly produced for ontology alignment systems. The fundamental thought is to clearly examine both the common substrings and non-common substring lengths of the two matched strings.

3.17 Soft Jaccard

Soft Jaccard, as against to Jaccard metric, is a set metric only which should be utilized together with a base matching technique. To begin with, the base matching technique is executed over every pair of words in both strings. The technique totals those pairs of words for which the base technique results in larger value as compared to a certain threshold. The total value is then divided by the quantity of words in the larger string. This is given in the following formula:

$$\text{SoftJaccard}(s1, s2, t) = \frac{|\text{sim}(A_i, B_i) \geq t|}{\max(|A|, |B|)}$$

A shows set of words in one string while B shows in the other string, t is the threshold and sim represent that base metric. The subscript i and j is the counter for words in the first and second strings respectively.

3.18 Substring Inclusion

This metric tells whether one string is part of the other or not.

3.19 Suffix

This metric tells whether one string is a suffix of the other string or not.

3.20 TF-IDF/cosine (5)

TF-IDF is an abbreviation of Term Frequency - Inverse Document Frequency. Information retrieval systems uses this method for indexing the document. The word frequency shows that in a document how many times a word shows up, divided by the total quantity of words in the document. The second part of the term is the inverse document frequency which is equal to the logarithm of the total number of documents divided by the quantity of documents in which the word in query exists.

4. EXPERIMENTAL SETUP

For comparing the performance of aforementioned metrics for the purpose of ontology alignment, various experiments have been conducted. These experiments implemented Java along with the Alignment API¹ and SecondString².

4.1 Alignment API

This is a java based API and is basically used to manipulate alignments of ontologies. It provides implementation of several interfaces and functions. It primarily implements three interfaces that are Alignment, Cell and Relation. *Alignment* describes specification of a particular alignment, *Cell* outline a correspondence in between entities and *Relation* does not require a specific feature. Besides these basic interfaces, certain others are also implemented in this API like *AlignmentProcess* which is utilized for implementing new matching metrics and *Evaluator* which compares an alignment with a reference alignment.

The alignment API provides certain functions like parsing an RDF/XML alignment, computing and comparing alignments, thresholding an alignment with a certain value as a threshold, and outputting an alignment in several formats like OWL, SWRL, XSLT and RDF.

¹ <http://alignapi.gforge.inria.fr>

² <http://secondstring.sourceforge.net>

4.2 Second String Library

This library was developed by a group of researchers in Cornegie Mellon University. It is a java based open-source library which implements a number of string matching techniques.

5. RESULTS AND ANALYSIS

The string metrics were evaluated with the above mentioned Java based libraries utilizing ontologies which are provided by alignment API package. For comparison, the concept of precision and recall is used. These concepts are taken from the domain of Information retrieval and are used to compare ontology matching techniques.

Precision and recall depend on the comparison of produced alignment A with an already available reference alignment R, successfully looking at which correspondences are found and which are certainly not found. These criteria are surely common and broadly acknowledged.

Precision values of various string metrics are given in Fig 1.

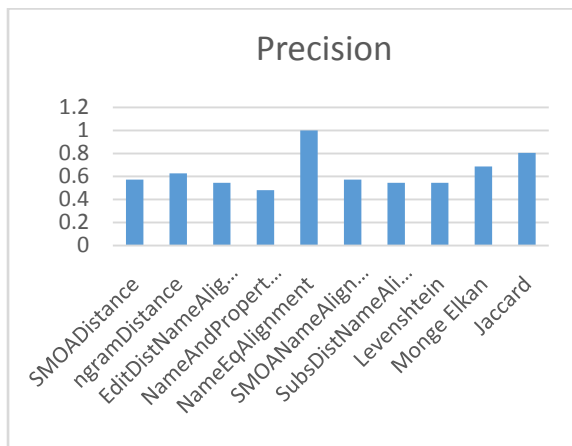


Figure 1. Precision values of String metrics

It can be noticed from the figure that with respect to precision, "NameEqAlignment" is the best and then Jaccard works better than others.

Recall of various string similarity metrics is given in the Fig 2 below.



Fig 2. Recall values of String Metrics

Unlike precision, NameEqAlignment is the worst with respect to recall values. With precision and recall only, one can't compare alignment techniques properly because a metric having high recall may have low precision and vice versa.

Therefore a new measure is introduced called F-measure which is equivalent to the harmonic mean of precision and recall i.e.

$$F = \frac{2PR}{P + R}$$

Where P represents precision and R represents recall. The F-measure values for the metrics is given in the Fig 3.

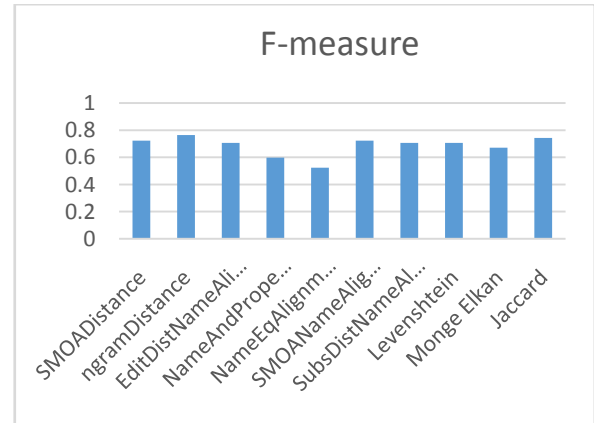


Fig 3. F-measure values of String metrics

NgramDistance and Jaccard work better in terms of F-measure and the NameEqAlignment measure works the worst.

The following graph shows the performance of every metric with respect to the three criteria mentioned above that are precision, recall and F-measure.

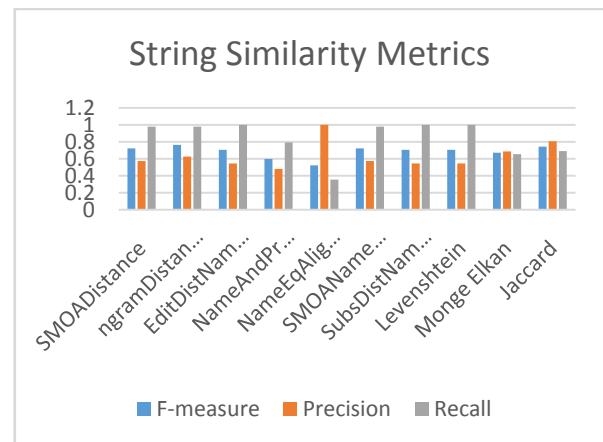


Fig 4. F-measure, Precision and Recall values of String Metrics

Further, several ontology matching systems were analysed which are using string matching techniques for aligning ontologies. These systems are discussed below:

5.1 CroMatcher

It is automated ontology matching system. In order to fully exploit all the information contained in the submitted ontologies, this system utilizes nine basic matchers. The basic matchers are from String matchers as well as Structure based matchers. Initially, a parallel arrangement of string based matchers are used and its outcome is then utilized by structure based matchers. Focus of this work is on string matchers only. CroMatcher uses ngramDistance similarity measure to compare IDs and annotations of entities belonging to the matching ontologies. To compare larger texts in ontologies it uses the TF/IDF and Cosine similarity measure. The result of these

measures is then passed to the structure based matchers for finding final alignment of input ontologies.

5.2 Agreement Maker Light (AML)

This system is primarily designed to efficiently align large sized ontologies. It implements various matchers like Lexical matcher, Mediating Matcher, Word Matcher and the Parametric String Matcher. Word Matcher and the Parametric String Matcher are discussed here. The Word matcher measures the string similarity between classes of input ontologies using Jaccard string similarity algorithm. AML uses Parametric String Matcher as a secondary matcher which implements TF/IDF and Cosine similarity algorithms.

5.3 RiMOM:

This system uses TF/IDF similarity metric in Data Processing phase and to find predicate alignment it uses Jaccard similarity matrix.

5.4 XMap:

This system is fast, efficient and scalable with a capacity to deal with hundreds of thousands entities. It works in several layers. It employs several string similarity metrics like Levenshtein, Jaro Winkler, n-gram and Jaccard distance to determine similarity in names, labels and comments contained in entities of input ontologies.

In OAEI 2016 the performance of the above mentioned ontology alignment systems are given in the table below

Table 1. Performance of matchers in 2016

Matching System	STRING methodology used	Precision	Recall	F-measure
AgreementMakerLight (AML)	Jaccard (as a word-based string similarity algorithm), TFIDF and cosine similarity algorithms (as parametric string matcher)	1.00	0.24	0.38
CroMatcher	Ngram(for comparing ID and annotation text), TF/IDF and cosine similarity for comparing larger texts.	0.96	0.83	0.89
RIMOM	TF-IDF and Jaccard	00	00	NaN
XMAP	Levenshtein distance, Jaro-Winkler distance, n-grams, Jaccard distance, the Cosine	0.95	0.40	0.56

Matchers with respective string methodologies are mentioned in the table. The data in the table is visualized as graph in Fig 5.

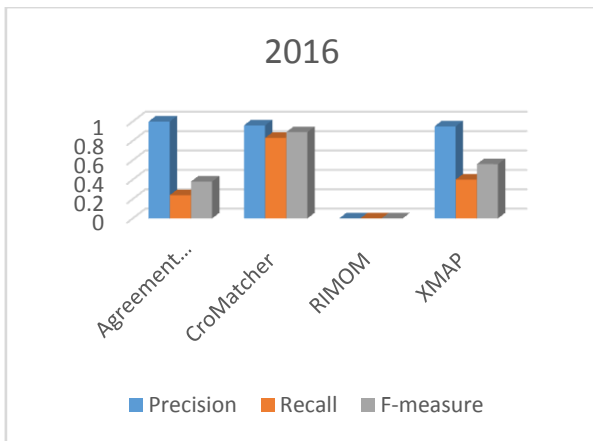


Fig 5. Performance of matchers in 2016

CroMatcher worked very well in every aspect. In terms of F-measure it resulted with the best figure i.e. 0.89. It uses first string-based matching metric whose result is then used by structure based matching metrics. If string-based metric is not used then only structure based metric can't give the desired results because string-based metric gives information about similarity of entity names, labels, comments and annotations.

AML which used Jaccard as a primary matching algorithm for determining word based similarity and TF/IDF and cosine similarity metrics as secondary. It results in best precision but low recall and F-measure.

RiMOM in 2016 did not produce valid results.

Xmap which uses several string similarity metrics also produces better results with respect to precision and F-measure. String metrics helps in finding correspondence between class names, comments and URIs. This information is then Semantic and structural layers. Thus string similarity metrics works as a starting point in every ontology matching system.

Similarly if performance of these systems in 2015 OAEI campaign was analysed, the following results can be obtained.

Table 2. Performance of matchers in 2015

Matching System	Precision	Recall	F-measure
AgreementMakerLight (AML)	0.92	0.39	0.55
CroMatcher	0.94	0.82	0.88
RiMOM	0.99	0.99	0.99
XMAP	1	0.40	0.57

Matchers with respective string methodologies are mentioned in the table. The data in the table is visualized as graph in Fig 6.

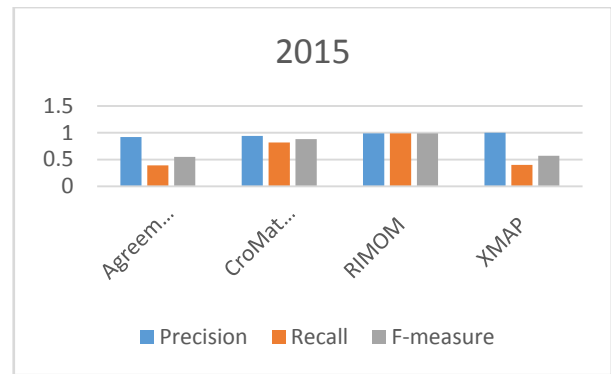


Fig 6. Performance of matchers in 2015

This year RiMOM produced the best results with F-measure value equal to 0.99. It uses TF/IDF and Jaccard in the first two phases of ontology alignment process. CroMatcher also worked well with F-measure equal to 0.88.

From the above analysis, it can be concluded that Jaccard in combination with TF/IDF similarity metric works good as well as ngram also better.

6. CONCLUSION

To connect the semantic space among various related ontologies, computer scientists and specialists in the field of Semantic Web have suggested and executed different ontology matching systems. In any case, these strategies require greater enhancement and continuous research should be made to get better values of precision, recall and F-measure for the alignment of ontologies. This work analysed various string similarity metrics for the task of ontology alignment. Definition of string metrics is taken from two libraries that are Alignment API and secondstring library. These metrics are used to align different ontologies provided with Alignment API package and then evaluated the performance of these metrics in terms of precision, recall and F-measure. Further, different ontology alignment systems are studied which used string similarity metrics in combination of structural similarity metrics and linguistic similarity metrics. The role of string similarity metrics is analysed and discussed here.

7. REFERENCES

- [1] McGuinness, Deborah L., and Frank Van Harmelen. "OWL web ontology language overview." *W3C recommendation* 10.10 (2004): 2004.
- [2] Gruber, Thomas R. "A translation approach to portable ontology specifications." *Knowledge acquisition* 5.2 (1993): 199-220.
- [3] Kutz, Oliver, Till Mossakowski, and Dominik Lücke. "Carnap, Goguen, and the hyperontologies: logical pluralism and heterogeneous structuring in ontology design." *Logica Universalis* 4.2 (2010): 255-333.
- [4] Fensel, Dieter, et al. "OIL: An ontology infrastructure for the semantic web." *IEEE intelligent systems* 16.2 (2001): 38-45.
- [5] Benjamins, V. Richard, et al. "Law and the semantic web, an introduction." *Law and the Semantic Web*. Springer Berlin Heidelberg, 2005. 1-17.
- [6] Beisswanger, Elena, et al. "BioTop: An upper domain ontology for the life sciences." *Applied Ontology* 3.4 (2008): 205-212.

- [7] Ding, Yihong, et al. "Generating ontologies via language components and ontology reuse." NLDB. Vol. 2007. 2007.
- [8] Elliman, Dave, and J. R. G. Pulido. "Visualizing ontology components through self-organizing maps." Information Visualisation, 2002. Proceedings. Sixth International Conference on. IEEE, 2002.
- [9] Pulido, J. R. G., et al. "Identifying ontology components from digital archives for the semantic web." IASTED Advances in Computer Science and Technology (ACST) (2006): 1-6.
- [10] Guarino, Nicola. "Some organizing principles for a unified top-level ontology." AAAI Spring Symposium on Ontological Engineering. 1997.
- [11] Ikeda, Mitsuru, Kazuhisa Seta, and Riichiro Mizoguchi. "Task ontology makes it easier to use authoring tools." IJCAI (1). 1997.
- [12] Kaiya, Haruhiko, and Motoshi Saeki. "Using domain ontology as domain knowledge for requirements elicitation." Requirements Engineering, 14th IEEE International Conference. IEEE, 2006.
- [13] Pires, Carlos Eduardo, et al. "SemMatcher: A Tool for Matching Ontology-based Schemas." 24th Brazilian Symposium on Data Bases (SBBD'09), Fortaleza, Brazil. 2009.