

An Iterative Automatic Final Alignment Method in the Ontology Matching System

Marko Gulić

*Faculty of Maritime Studies Rijeka
University of Rijeka, Rijeka, Croatia*

marko.gulic@pfri.hr

Boris Vrdoljak

*Faculty of Electrical Engineering and Computing
University of Zagreb, Zagreb, Croatia*

boris.vrdoljak@fer.hr

Marin Vuković

*Faculty of Electrical Engineering and Computing
University of Zagreb, Zagreb, Croatia*

marin.vukovic@fer.hr

Abstract

Ontology matching plays an important role in the integration of heterogeneous data sources that are described by ontologies. In order to determine correspondences between ontologies, a set of matchers can be used. After the execution of these matchers and the aggregation of the results obtained by these matchers, a final alignment method is executed in order to select appropriate correspondences between entities of compared ontologies. The final alignment method is an important part of the ontology matching process because it directly determines the output result of this process. In this paper we improve our iterative final alignment method by introducing an automatic adjustment of final alignment threshold as well as a new rule for determining false correspondences with similarity values greater than adjusted threshold. An evaluation of the method is performed on the test ontologies of the OAEI evaluation contest and a comparison with other final alignment methods is given.

Keywords: automatic matching system, iterative final alignment, ontology matching, heterogeneous data integration

1. Introduction

A rapid development of information and communications technology has led to a huge increase in the amount of available data. Consequently, there is a large number of new data sources that describe the same domain of interest. As this data sources have been designed independently of each other, they are mutually heterogeneous and therefore difficult to interconnect. An ontology enriches a knowledge about some data source by describing the entities of this data source and relations between these entities in detail. Therefore, the use of ontologies facilitates the integration of heterogeneous data

sources with the same domain of interest. In computer science, ontology is a formal, explicit specification of a shared conceptualization [1]. Each ontology is expressed by using an ontology language that defines ontology entities and relations between them in a way comprehensible to a computer. One of the most popular ontology languages is Web Ontology Language (OWL) [2]. OWL is recommended by W3C (World Wide Web Consortium) [3] as an international standard for ontology representation. Ontology matching is the process of finding correspondences between entities of different ontologies [4]. Ontology matching is a key issue in the process of integrating heterogeneous data sources described by ontologies [4].

There is a large number of applications (Web navigation and searching, information integration, ontology engineering...) where ontology matching can help in using data from different sources [4]. The use of ontology matching can be explained in the example of search for information on the World Wide Web. An important task of the state-of-the-art search engines is to determine the user's intent, which is "hidden" within the query [5]. One of the solutions to the problem of inferring the intent of the query is to introduce ontologies as additional semantic components of webpages. Ontologies will support machines in grasping the understanding of the data and thus the user will be enabled with a more efficacious search of the Web, including an automated matching of different webpages.

In order to automate the ontology matching process, an ontology matching system has to be developed [4]. Ontology matching system usually consists of several basic matchers which, using the information of the compared ontologies, determine correspondences between entities of compared ontologies. The results of basic matchers are then aggregated in order to obtain best possible correspondences between ontologies. Before the ontology matching process is finished, the final alignment method is executed. This method selects those correspondences whose entities represent the same real-world object within compared ontologies. A quality final alignment method is crucial for the ontology matching process. If the final alignment method does not select correct correspondences in the final alignment which is the final output of an ontology matching process, then the system would not perform the quality matching process for any pair of compared ontologies.

As each ontology is unique, the final alignment method needs to adjust its parameters' values that will be set optimally for determining correspondences between current pair of compared ontologies. It is difficult for an expert to adjust these optimal parameters' values and even more for non-expert user. Therefore, an automatic adjustment of parameters' values would facilitate the usage of an ontology matching system especially for non-expert users. The focus of this paper is on automatic adjustment of the final alignment method parameters, especially the adjustment of the correspondence threshold that determines the minimal value for correspondences that will be part of the final alignment. The Threshold final alignment method is an antecedent of today's final alignment methods and is often integrated in these methods. The adjustment of the threshold value is usually determined manually within matching systems considering the experience of developers, therefore it is not necessarily adjusted in an optimal way for each pair of compared ontologies. Fully automating the final alignment method would

significantly improve the process of determining the correspondences between any pair of ontologies.

In this paper our contribution is twofold. First, we improved our final alignment method presented in [6] by introducing the automatic adjustment of correspondence threshold value instead of manual adjustment which was performed by a user of our ontology matching system. Only the correspondences that have the highest value for both ontology entities (with respect to all entities from the other ontology) and have the similarity value greater than automatically adjusted correspondence threshold enter the final alignment.

Second, we propose a new restriction rule of discarding previously selected correspondences with highest value between two entities from the final alignment if there is no significant difference between the values of the highest and the second highest correspondence of one of the compared entities. The second highest correspondence of an entity has the second highest value considering all correspondences between this entity and any entity of the other ontology.

The new final alignment method was tested on the *Benchmark* ontology track, which is the best known ontology matching test case of the evaluation organized by the Ontology Alignment Evaluation Initiative (OAEI) [7], [8].

The paper is organized as follows. In Section 2 basic terminology of ontology matching and the main components are introduced. In Section 3 we discuss related work. In Section 4 our final alignment method for automatically determining the appropriate correspondences between entities of compared ontologies is presented. In Section 5 the evaluation of our method is presented. Finally, the conclusion is given in section 6.

2. Ontology Matching

2.1. Terminology

In this subsection, the basic terms referring to ontology matching, adopted from [4], [6], are presented.

As stated before, **ontology matching** is the process of finding semantic relationships or correspondences between entities of different ontologies. Ontology matching is defined as function:

$$A = f(O, O', p, r) \quad (1)$$

Alignment A is the matching result between ontologies O and O' , p is a set of parameters within the matching process, and r is a set of resources used in the matching process.

Correspondence is a similarity value between two entities of different ontologies. A correspondence is defined as:

$$c(e_i, e'_j) = n \quad (2)$$

Index e_i is an entity of the ontology O , e'_j is an entity of the ontology O' , and n is a real number from the interval $[0, 1]$. The higher the correspondence, the greater the similarity between two entities.

Alignment A is a set of all correspondences $c(e_i, e'_j)$ between entities e_i of ontology O and entities e'_j of ontology O' that are found in the matching process. Alignment is actually the final output of an ontology matching process.

2.2. Ontology matching system

A large number of ontology matching systems exists. OAEI evaluation [7], [8], the most authoritative evaluation of ontology matching systems, performs evaluation on at least 15 new or modified matching systems every year. Although each system is unique, the matching systems usually consist of similar components. In general, the matching process can be divided into three main types of components [4], [6]: basic matchers, aggregation method for basic matchers' results, and final alignment method.

Each basic matcher determines correspondences between entities of compared ontologies by using information from one or more ontology components. Consequently, an ontology matching system usually consists of several matchers in order to utilize all information from compared ontologies and to improve matching results. Basic matchers are generally divided into element and structure matchers [4]. Element matchers determine correspondences just by analyzing components of compared entities, while structure matchers determine correspondences by analyzing the components of compared entities and also the ontology structure i.e. the relations of these entities with other entities.

Basic matchers are mutually connected (the most known compositions of basic matchers are sequential and parallel composition [4]) and their results are aggregated for determining high quality correspondences between entities.

After the aggregation of basic matchers' results, a final alignment method has to select correct correspondences between entities of compared ontologies for the final alignment A .

3. Related Work

Final alignment is the one of the most important subprocesses within the matching process. As stated before, it is the output of the ontology matching process and if a final alignment method does not perform a quality selection of correspondences between ontologies, the whole matching process will achieve poor results.

In this section a number of final alignment methods that are used in various ontology matching system are described and analyzed.

Threshold method was presented in the Coma system [9], [10] for the first time, and is also used in the actual systems described in [11], [12], [13], [14], [15], [16], [17] as part of their composite final alignment methods. According to this simple method, only the correspondences that have a value greater than the given threshold are selected in the final alignment. The problem of using only this method for the final alignment process is reflected on the adjustment of optimal threshold value. If the

threshold value is too high, a small number of correspondences will be selected in the final alignment, therefore it can happen that certain correct correspondences will not be selected in the final alignment. As opposed to high threshold value, if the threshold value is too low, a large number of correspondences will be selected in the final alignment. Consequently, more than one correspondence that contains the same entity can be selected in the final alignment. As the relationship between entities of compared ontologies is usually in 1:1 ratio (one entity of first ontology corresponds to one entity of second ontology), a too low threshold value results in a large number of false correspondences within the final alignment. Hence, it is difficult for an expert to adjust the optimal threshold value for each pair of compared ontologies and even more for an ordinary user. Therefore, an automatic adjustment of the threshold value would significantly improve the final alignment process and facilitate the usage of a matching system. In this paper, we propose the automatic adjustment of threshold value within our new version of the final alignment method.

The **MaxN** method was also presented in the Coma system [9], [10]. It takes N greatest correspondences of each entity in the final alignment. When N is greater than one, The MaxN method has the same problem as Threshold method and there is a possibility of selecting more than one correspondence of the same entity in the final alignment. Therefore, the ontology matching systems usually use Max1 method for final alignment process. The Max1 method selects in the final alignment only one correspondence of each entity that has the highest value among all correspondences of this entity. The example of using Max1 method inside the matching systems can be seen in [16], [18]. The deficiency of this method is that it does not iteratively select the correspondences in the final alignment. For example, let the same hierarchy structure of three entities exists within the ontologies that have to be matched. One entity is the parent of the other two entities. A correct set of correspondences between compared ontologies would contain one correspondence between parent entities and two correspondences between children entities. The parent entity of the second ontology is much better described than its children entities, therefore all entities of the first ontology have the greatest correspondence with the same (parent) entity of the second ontology. As the correspondence value between parent entities has the highest value, it enters the final alignment. The correspondences between children entities of the first ontology and the parent entity of the second ontology do not enter the final alignment because this correspondences are not the greatest correspondences of the parent entity of second ontology. Also, the correspondences between children entities of the first ontology and children entities of the second ontology do not enter the final alignment because children entity of the first ontology have the greatest correspondences with the parent entity of the second ontology as stated before. This problem could be resolved by iterative selection of correspondences that enter the final alignment. In first iteration, the correspondence between parent entities would enter the final alignment. Before the start of the second iteration, all correspondences that contain the parent entity of the first ontology or the parent entity of the second ontology would be removed from the pool of possible correspondences. Thus, there would be the possibility of selecting the correspondences between two children in the final alignment because these correspondences now can be the highest

correspondences within the second iteration. We resolve this problem within our final alignment method by introducing the iterative selection of correspondences during the final alignment process.

The **MaxDelta** method was initially presented in [9], [10]. In final alignment, it selects only the correspondence with the highest value and all correspondences with a value differing at most by a tolerance value *delta* with regard to the highest value. The MaxDelta method has similar problems as the Threshold method. Using the MaxDelta method, more than one correspondence of the same entity can be selected in the final alignment. Also, there exists a problem of setting optimal delta value for final alignment process.

The **MaxPercent** method [15] is similar to the MaxDelta method. Here, only the correspondence with the highest value and all correspondences with a value differing at most by a tolerance *percent* value with regard to the highest value are selected in the final alignment.

Four **filter methods** are used in [12] for selecting the appropriate correspondences in the final alignment. First method discards all obtained correspondences between entities in which at least one of the entities is not the part of compared ontologies (external entity). The external entities are usually used in the matching process for more comprehensive description of the entities that are contained within compared ontologies. Second method discards all obtained correspondences between data properties that have defined ranges with different data types. The problem of this method is a strict discharge of all correspondences with different ranges. For example, the correspondence value between two data properties could be very high considering the results obtained by previously executed basic matchers. The first data property has range defined as integer and the second data property has range defined as positive integer. Although the data types are very similar and the correspondence value is high, this correspondence will be discarded given the second method. Therefore, second method has to be more flexible when comparing data properties. The remaining methods select the correspondences that will enter the final alignment. For each entity of the first ontology, only the correspondence that has the highest similarity value with certain entity of the second ontology is selected for the final alignment. The same process is also performed for each entity of the second ontology. Here, the problem occurs when two entities of the first ontology have the greatest similarity with the same entity of the second ontology and thus the false correspondences can enter the final alignment. In the end, the fourth filter method, which is actually the threshold method, is executed. The authors did not explain the determination of the optimal threshold value.

In [14], the final alignment method **combines** two methods: **Threshold** and **Max1**. Also, there is an additional rule of unchangeable order of basic matchers' importance and this order rule influences on the final alignment process. Hence, the basic matchers are ordered unchangeably by their importance determined by the authors' experience. First, all correspondences within the results of the most important basic matcher are selected in the final alignment if they satisfy the conditions of Threshold and Max1 together. Second, the same process is repeated for the second most important basic matcher without taking into consideration the selection of

correspondences that contains entities which are included in selected correspondences of the first iteration. The process is repeated until the selection of correspondences from the last basic matcher. Here, the problem occurs when the correspondence of an entity of one ontology enters the final alignment according to the results of highly important basic matcher, and there exists better correspondence of this entity within the results of lowly important basic matcher. This better correspondence will not be taken into consideration of entering the final alignment according to the unchangeable importance order of basic matchers. For example, it can occur that the highly important basic matcher achieved poor results because the ontology components (e.g. labels and comments of entities), from which this basic matcher uses the information to determine correspondence value between two entities, are not implemented within compared ontologies. Therefore, if some basic matcher is ordered highly according to its importance and if the order is always unchangeable, there is a possibility of selecting false correspondences in the final alignment.

In the matching system described in [19], [20] the authors propose the usage of **two** additional **rules** together with **Threshold** method for determining the final alignment. These rules restrict the selection of the correspondences obtained by the structural basic matchers in the final alignment. First rule examines the quality of correspondences obtained by element and structure basic matcher. If the quality of the correspondences obtained by element matchers is high and the quality of the correspondences obtained by structure matchers is also high, all correspondences obtained by element matchers and only the highest quality correspondences obtained by structure matchers that satisfy the threshold value will enter the final alignment. The authors did not explain the method for determining the quality of a correspondence. The second rule also examines the quality of obtained correspondences. If the quality of the correspondences obtained by element matchers is low and the quality of the correspondences obtained by structure matchers is high, all correspondences obtained by structure matchers that satisfy the threshold value, enter the final alignment. These two rules decrease the possibility of selection more than one correspondence of the same entity in the final alignment using the Threshold method, but the problem still exists.

The authors of the ontology matching system described in [21], [22], [23], [24] propose the final alignment method based on the **Hungarian** method [25]. The Hungarian method is an optimization algorithm that finds the maximum sum of correspondences' values within the matrix of correspondences (each entity from the first ontology is related with each entity from the second ontology) in a way that each entity of compared ontologies is included only in one selected correspondence of all correspondences that form this maximum sum. Considering the correspondences that form the maximum sum of correspondences' value satisfying the condition of including only one correspondence of each entity within these correspondences, only the correspondences that have value greater than the defined threshold value (**Threshold** final alignment method) will enter the final alignment. As stated before, the relationship between entities of compared ontologies is usually in 1:1, therefore the Hungarian method satisfies this condition by restricting the number of correspondences that contain the same entity to one. This method resolves the problem

of entering multiple correspondences of the same entity in the final alignment that occurs by using the final alignments methods: Threshold, MaxDelta and MaxPercent. The main drawback of this method is the possibility of exclusion the correspondence with the highest similarity value from the final alignment given that the method searches for the maximum sum of correspondences' values, not the highest value of each correspondence. For example, a 2x2 matrix contains correspondences as follows: $c(e_1, e_1') = 0.95$, $c(e_1, e_2') = 0.90$, $c(e_2, e_1') = 0.90$ and $c(e_2, e_2') = 0.80$. Correspondences $c(e_1, e_2')$ and $c(e_2, e_1')$ will enter the final alignment because the sum of these two correspondences is maximal for this matrix of correspondences (1.80). Although the correspondence $c(e_1, e_1')$ has the highest value (0.95) and is obviously the correct correspondence, it is not selected in the final alignment. Furthermore, the problem of adjusting the optimal threshold value also exists. In our new method, the optimal threshold value will be automatically adjusted.

In [6], we proposed an **iterative** final alignment method that resolves problems detected in previously described final alignment methods. This method allows only one correspondence of an entity to be included in the final alignment like in Max1 and Hungarian methods. This correspondence had to be the correspondence with the highest value for both ontology entities that are related through this correspondence. Hence, this method always includes the greatest correspondence of all obtained correspondences in the final alignment and resolves the problem occurring in Hungarian method where there is a possibility of not including the greatest correspondence in the final alignment. Iterative process of the final alignment resolves the problem of Max1 when multiple entities of one ontology have the greatest correspondence with the same entity of another ontology, therefore correspondence of only one of these entities related to this same entity is included in the final alignment. The iterative process of the final alignment allows that a correspondence which is not the greatest correspondence of an entity can enter the final alignment if none of the remaining correspondences that contain this entity has not been included in the final alignment yet. Furthermore, just the correspondences that have value greater than the defined threshold are included in the final alignment. This threshold always has a low value because the selected highest correspondences are usually correct. However, this low threshold value sometimes can result with mistake. In this paper, we propose a new final alignment method that is based on the method described in [6]. The process of an automatic adjustment of the optimal threshold value for currently compared ontologies is introduced to resolve the problem of low threshold value described above. We also propose a better way of determining the correct highest correspondences whose entities represent the same object in a real-world.

4. A New Final Alignment Method

As stated before, in this paper we improve our final alignment method [6] by introducing an automatic adjustment of final alignment threshold as well as a new rule for determining false correspondences with similarity values greater than adjusted threshold. The method described in [6] is based on the highest correspondences found in the aggregated alignment results of basic matchers. A correspondence between

entity e_i of ontology O and e'_j of ontology O' is the highest correspondence if and only if it has value higher than any other correspondence of either e_i or e'_j with some other entity. Hence, only the highest correspondences that have the confidence value higher than threshold enter the final alignment. We assume that this method is very reliable because it includes only the highest correspondences between two entities in both directions of mapping in the final alignment.

The first drawback of the method described above is that the threshold value has to be determined manually. The method takes into final alignment only the highest correspondences as the most relevant, because these correspondences have the highest value for both ontology entities that are related through this correspondence. Therefore, the threshold value is always low. However, the final alignment sometimes contains a small number of highest correspondences that have a value that is low, but still greater than the threshold, and these correspondences are not necessarily correct although they are the highest for a certain pair of ontologies. This problem can be solved by using statistical analysis. We propose that the threshold value is set to the arithmetic mean of all found highest correspondences diminished by its standard deviation. Thus, all the highest correspondences in the alignment (i.e. the matrix of all correspondences) are determined, and then the arithmetic mean and standard deviation of their values are determined. Finally, the final alignment threshold is set to the value of arithmetic mean diminished by the value of its standard deviation. This way, final alignment will contain most of the highest correspondences, except for those with the lowest values. Furthermore, the threshold value will depend on the values of highest correspondences between entities of different ontologies. If a large number of highest correspondences have a high (low) value, the threshold value will be high (low). Therefore, the threshold value is automatically adjusted to the results of the highest correspondences of the matching ontologies. As stated before, the highest correspondences are the most reliable, because the highest correspondence between entities e_i and e'_j has the value higher than any other correspondence of either e_i or e'_j with some other entity. Figure 1. shows an example of determining the final alignment threshold on the alignment matrix. The pseudocode for determining the final alignment threshold is given in Algorithms 1 and 2.

The threshold is determined by finding the highest correspondences in the final alignment matrix first (Algorithm 2, line 1 - R – a common alignment, line 4 - $H \leftarrow \text{determineHighSecondHighCorr}(R, 0, 0)$). The values of thresholds for finding the highest correspondences (Algorithm 1, line 1 - thr_{HC} and thr_{SEC}) are set to 0, because for calculating the threshold value all highest correspondences are used in this calculation regardless of the values of the highest correspondences of entities and their second highest correspondences, which is important when determining the correspondences that enter the final alignment. In the first row there is the highest correspondence $c(e_1, e'_5) = 0.29$ (Algorithm 1, line 38 – $hMax_{15} = \text{TRUE}$, $vMax_{15} = \text{TRUE}$, $p_{15} > 0$) because this correspondence has the highest value out of all correspondences of the entity e_1 (Algorithm 1, line 8-20 – row:1, $maxIndex = 5$, $hMax_{15} = \text{TRUE}$, $maxValue = 0.29$) and the entity e'_5 (Algorithm 1, line 21-33 – column:5, $maxIndex = 1$, $vMax_{15} = \text{TRUE}$, $maxValue = 0.29$). The same way, the

remaining highest correspondences in the matrix of all correspondences are determined: $c(e_2, e_2') = 0.55$, $c(e_3, e_1') = 0.81$, $c(e_4, e_3') = 0.75$ and $c(e_5, e_4') = 0.55$.

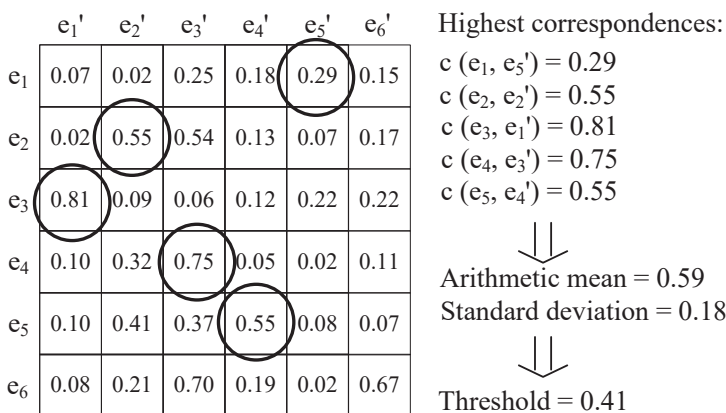


Figure 1. Example of determining the final alignment threshold

The correspondence $c(e_6, e_3') = 0.70$ is not the highest correspondence (Algorithm 1, line 38 – $hMax_{63} = \text{TRUE}$, $vMax_{63} = \text{FALSE}$, $p_{63} > 0$), because there exists a correspondence of the entity e_3' that is greater than 0.70 (Algorithm 1, line 21-33 – column:3, $maxIndex = 4$, $vMax_{43} = \text{TRUE}$, $maxValue = 0.75$). Also, the correspondence $c(e_6, e_6') = 0.67$ is not the highest correspondence (Algorithm 1, line 38 – $hMax_{66} = \text{FALSE}$, $vMax_{66} = \text{TRUE}$, $p_{66} > 0$), because the entity e_6 has a higher correspondence with the entity e_3' - the value is 0.70. After determining the highest correspondences (Algorithm 2, line 4), the arithmetic mean and standard deviation should be calculated (Algorithm 2, line 5-15). Arithmetic mean of the values of highest correspondences is 0.59 (Algorithm 2, line 7-10 – $aritMean = 0.59$), and the standard deviation is 0.18 (Algorithm 2, line 11-15 – $std = 0.18$). Therefore, the threshold is set to 0.41 (Algorithm 2, line 16 – $thr_{FIN} = aritMean - std = 0.41$). This way the highest correspondences with low values will surely not enter the final alignment. In this example, $c(e_1, e_5') = 0.29$ is such a correspondence.

After determining the threshold, the iterative process of selecting correspondences for the final alignment starts. The process was explained in detail in [6]. Its drawback is that the highest correspondence between two entities e_i , and e_j' is not compared with the second highest correspondence of either e_i or e_j' . It can happen that the second highest correspondence is almost equal to the highest correspondence. It can be concluded that in this case the highest correspondence is not as reliable as it seemed at first glance.

Procedure `determineHighSecondHighCorr`

- 1 Data: P – an alignment matrix of dimension $|O| \times |O'|$; thr_{HC} – highest correspondence threshold; thr_{SEC} - minimal difference threshold between the highest and the second highest correspondence value of an entity

```

2  Result:  $H$  – a set of highest correspondences
3   $H \leftarrow \emptyset$ ;  $HMax \leftarrow Matrix(|O|, |O'|)$ ;  $VMax \leftarrow Matrix(|O|, |O'|)$ ;  $HSecond \leftarrow Array(|O|)$ ;  $VSecond$ 
    $\leftarrow Array(|O'|)$ ;
4  for  $i \leftarrow 1$  to  $|O|$  do
5     $hSecond_i \leftarrow 0$ ;
6    for  $j \leftarrow 1$  to  $|O'|$  do
7       $hMax_{ij} \leftarrow FALSE$ ;  $vMax_{ij} \leftarrow FALSE$ ;  $vSecond_j \leftarrow 0$ ;
8  for  $i \leftarrow 1$  to  $|O|$  do
9     $maxVal \leftarrow -1$ ;  $maxIndex \leftarrow 0$ ;  $secondVal \leftarrow -1$ ;  $secondIndex \leftarrow 0$ ;
10   for  $j \leftarrow 1$  to  $|O'|$  do
11     if  $p_{ij} > maxValue$  then
12       if  $maxIndex > 0$  then
13          $secondIndex \leftarrow maxIndex$ ;
14          $secondValue \leftarrow maxValue$ ;  $maxValue \leftarrow p_{ij}$ ;  $maxIndex \leftarrow j$ ;
15     else if  $p_{ij} > secondValue$  and  $p_{ij} < maxValue$  then
16        $secondValue \leftarrow p_{ij}$ ;  $secondIndex \leftarrow j$ ;
17     else if  $p_{ij} = maxValue$  then
18        $maxIndex \leftarrow 0$ ;  $secondValue \leftarrow p_{ij}$ ;  $secondIndex \leftarrow j$ ;
19   if  $maxIndex > 0$  then
20      $hMax_{i,maxIndex} \leftarrow TRUE$ ;  $hSecond_i \leftarrow secondIndex$ ;
21   for  $j \leftarrow 1$  to  $|O'|$  do
22      $maxVal \leftarrow -1$ ;  $maxIndex \leftarrow 0$ ;  $secondVal \leftarrow -1$ ;  $secondIndex \leftarrow 0$ ;
23     for  $i \leftarrow 1$  to  $|O|$  do
24       if  $p_{ij} > maxValue$  then
25         if  $maxIndex > 0$  then
26            $secondIndex \leftarrow maxIndex$ ;
27            $secondValue \leftarrow maxValue$ ;  $maxValue \leftarrow p_{ij}$ ;  $maxIndex \leftarrow i$ ;
28       else if  $p_{ij} > secondValue$  and  $p_{ij} < maxValue$  then
29          $secondValue \leftarrow p_{ij}$ ;  $secondIndex \leftarrow i$ ;
30       else if  $p_{ij} = maxValue$  then
31          $maxIndex \leftarrow 0$ ;  $secondValue \leftarrow p_{ij}$ ;  $secondIndex \leftarrow i$ ;
32     if  $maxIndex > 0$  then
33        $vMax_{maxIndex,j} \leftarrow TRUE$ ;  $vSecond_j \leftarrow secondIndex$ ;
34   for  $i \leftarrow 1$  to  $|O|$  do
35      $indexSecondRow = hSecond_i$ ;
36     for  $j \leftarrow 1$  to  $|O'|$  do
37        $indexSecondColumn = vSecond_j$ ;
38     if  $hMax_{ij}$  and  $vMax_{ij}$  and  $p_{ij} > thr_{HC}$  and  $(p_{ij} - p_{i, indexSecondRow}) > thr_{SEC}$  and  $(p_{ij} - p_{indexSecondColumn, j}) > thr_{SEC}$  then
39        $h.row \leftarrow i$ ;  $h.col \leftarrow j$ ;  $H \leftarrow H \cup \{h\}$ 

```

Algorithm 1. Procedure determineHighSecondHighCorr, used by algorithm for determining the final alignment threshold value (Algorithm 2) as well as the algorithm that produces the final alignment (Algorithm 3)

For instance, let there exist the highest correspondence $c(e_2, e_2') = 0.55$ and other highest correspondences of the entity $e_2 - c(e_2, e_3') = 0.54$ and the entity $e_2' - c(e_3, e_2') = 0.41$. The correspondence $c(e_2, e_2')$ is the highest correspondence, but the correspondence $c(e_2, e_3')$ as the second highest correspondence of the entity e_2 is less than $c(e_2, e_2')$ only by 0.01. We suppose that in this case the correspondence $c(e_2, e_2')$ is not so reliable as it would be in case when, for instance, the difference between the first and the second correspondence is 0.05. Therefore, we introduce a new restriction rule that will eliminate from the final alignment the highest correspondence between entities e_i and e_j' where there exists the second highest correspondence of the entity e_i or the entity e_j' , whose value is almost equal to the highest correspondence. This way there may be a smaller number of correspondences in the final alignment, but on the other hand, they will be more reliable.

Procedure `determineFinAlignThreshValue`

```

1  Data:  $R$  – a common alignment of dimension  $|O| \times |O'|$ ;
2  Result:  $thr_{FIN}$  – final alignment threshold;
3   $H \leftarrow \emptyset$ ;  $thr_{FIN} \leftarrow 0$ ;
4   $H \leftarrow \text{determineHighSecondHighCorr}(R, 0, 0)$ ;
5  if  $H \neq \emptyset$  than
6     $count \leftarrow 0$ ;  $aritMean \leftarrow 0$ ;  $std \leftarrow 0$ ;
7    foreach  $h \in H_k$ 
8       $i \leftarrow h.row$ ;  $j \leftarrow h.col$ ;
9       $count \leftarrow count + 1$ ;  $aritMean \leftarrow aritMean + r_{ij}$ ;
10    $aritMean \leftarrow aritMean / count$ ;
11   foreach  $h \in H_k$ 
12      $i \leftarrow h.row$ ;  $j \leftarrow h.col$ ;
13      $std \leftarrow std + (r_{ij} - aritMean)^2$ ;
14    $std \leftarrow std / count$ ;
15    $std \leftarrow \text{squareRoot}(std)$ ;
16    $thr_{FIN} \leftarrow aritMean - std$ ;

```

Algorithm 2. Algorithm for determining the final alignment threshold value used by algorithm that produces the final alignment (Algorithm 3)

Figure 2. shows an example of iterative selection of correspondences for final alignment (Threshold value = 0.41) by introducing a new restriction rule (e.g. `ThresholdSecond = 0.02`) that will eliminate from final alignment all highest correspondences whose values are less than the defined `Threshold` and whose entities have at least one more correspondence that differs from the highest correspondence by less than 0.02. The pseudocode for determining the final alignment is given in Algorithm 3.

In the first iteration the highest correspondences $c(e_3, e_1') = 0.81$, $c(e_4, e_3') = 0.75$ and $c(e_5, e_4') = 0.55$ are found as they satisfy both `Threshold` and `ThresholdSecond`, defined in the process of determining the final alignment (Example for determining

$c(e_3, e_1') = 0.81 \rightarrow$ Algorithm 1, line 1 – $thr_{HC} = 0.41$, $thr_{SEC} = 0.02$, lines 4-20 – row 3: $maxIndex = 1$, $hMax_{31} = TRUE$, $maxValue = 0.81$, $secondValue = 0.22$, $secondIndex = 5$, $hSecond_3 = 5$; lines 21-33 - column: 1, $maxIndex = 3$, $vMax_{31} = TRUE$, $maxValue = 0.81$, $secondValue = 0.10$, $secondIndex = 4$, $vSecond_1 = 4$; line 38 – $hMax_{31} = TRUE$, $vMax_{31} = TRUE$, $p_{31} > thr_{HC}$, $(p_{31} - p_{35}) > thr_{SEC}$, $(p_{31} - p_{41}) > thr_{SEC}$.

First iteration

	e_1'	e_2'	e_3'	e_4'	e_5'	e_6'
e_1	0.07	0.02	0.25	0.18	0.29	0.15
e_2	0.02	0.54	0.54	0.13	0.07	0.17
e_3	0.81	0.09	0.06	0.12	0.22	0.22
e_4	0.10	0.32	0.75	0.05	0.02	0.11
e_5	0.10	0.41	0.37	0.55	0.08	0.07
e_6	0.08	0.21	0.70	0.19	0.02	0.67

Threshold = 0.41
ThresholdSecond = 0.02

Highest correspondences:

- $c(e_3, e_1') = 0.81$
- $c(e_4, e_3') = 0.75$
- $c(e_5, e_4') = 0.55$

~~$c(e_1, e_3') = 0.29$~~

Threshold!!

~~$c(e_2, e_3') = 0.54$~~

Second highest correspondence!!

$c(e_2, e_3') = 0.54$

Second iteration

	e_2'	e_5'	e_6'
e_1	0.02	0.29	0.15
e_6	0.21	0.02	0.67

Highest correspondences:
 $c(e_6, e_6') = 0.67$

Final alignment

- (e_3, e_1') , (e_4, e_3') ,
- (e_5, e_4') , (e_6, e_6')

Figure 2. Example of calculating the final alignment

The correspondence $c(e_2, e_2')$ (line 38 – $hMax_{22} = TRUE$, $vMax_{22} = TRUE$, $p_{22} > thr_{HC}$) does not enter the final alignment, because there is the correspondence between entities e_2 i e_3' (Algorithm 1, lines 4-20 – row 2: $secondValue = 0.54$, $secondIndex = 3$, $hSecond_2 = 3$) with the value 0.54, so the difference between the first ($c(e_2, e_2')$) and the second ($c(e_2, e_3')$) highest correspondence is 0.01 (Algorithm 1, line 38 – $(p_{22} - p_{23}) > thr_{SEC} \rightarrow FALSE$), which is less than the value of the ThresholdSecond (set to 0.02). Furthermore, the correspondence $c(e_1, e_5')$ with the value 0.29 does not enter the final alignment neither, because its value is less than the defined Threshold (Algorithm 1, line 38 – $p_{15} > thr_{HC} \rightarrow FALSE$). In the second iteration we take into account only those correspondences that contain entities that are currently not included in any correspondence within the final alignment or were not a part of any correspondence that is discarded according to the ThresholdSecond rule. All correspondences that contain at least one entity that is already included into the final alignment are removed from the alignment matrix [6] in the following iterations (Algorithm 3, lines 9-10). In this step the highest correspondence that is found is $c(e_6, e_6') = 0.67$. If this method had not been performed iteratively, this correspondence would not have entered the final alignment, because in the first step there is the

correspondence $c(e_6, e_3')$ with a value greater than the value of $c(e_6, e_6')$. As the entity e_3' has the highest correspondence with the entity e_4 , in the next step all the entities that already are in the final alignment are not taken into account, so the entity e_6 has a chance to connect with some other entity for the final alignment. The final alignment consists of four correspondences: $c(e_3, e_1')$, $c(e_4, e_3')$, $c(e_5, e_4')$ and $c(e_6, e_6')$.

Procedure `determineFinAlignThreshValue`

- 1 Data: R – a common alignment of dimension $|O| \times |O'|$; thr_{SEC} – minimal difference threshold between the highest and the second highest correspondence value of an entity
- 2 Result: F – a set of correspondences comprising the final alignment
- 3 $F \leftarrow \emptyset$;
- 4 $thr_{FIN} \leftarrow \text{determineFinAlignThreshValue}(R)$;
- 5 $H \leftarrow \text{determineHighSecondHighCorr}(R, thr_{FIN}, thr_{SEC})$;
- 6 while $H \neq \emptyset$ do
- 7 foreach $h \in H_k$
- 8 $F \leftarrow F \cup \{h\}$; $b \leftarrow h.row$; $c \leftarrow h.col$;
- 9 for $i \leftarrow 1$ to $|O|$ do $r_{ic} \leftarrow 0$
- 10 for $j \leftarrow 1$ to $|O'|$ do $r_{bj} \leftarrow 0$
- 11 $H \leftarrow \text{determineHighSecondHighCorr}(R, thr_{FIN}, thr_{SEC})$;

Algorithm 3. Algorithm for determining the final alignment

5. Evaluation

The evaluation was performed on *Benchmark biblio* ontology track [26] that is the largest test set in the ontology matching system evaluation organized by OAEI (Ontology Alignment Evaluation Initiative) [7], [8]. The *Benchmark biblio* test case contains more than 100 pairs of ontologies (written in OWL) and the alignment results between them. In each pair of ontologies, the first ontology contains all the information related to a specific domain. In the second ontology certain information (labels and comments of entities, ontology structure, properties etc.) are missing in order to test matching systems and the final alignment methods as part of these systems. Thus, the advantages and disadvantages of each matching system can be examined according to the missing component in a particular test. A new iterative final alignment method will be evaluated as a part of our CroMatcher system [6] which produced the best matching results for *Benchmark biblio* test case in OAEI 2016 [27]. Also, other final alignment methods that will be compared with our new method will be integrated as a part of CroMatcher system in order to test all methods in the same environment. The evaluation measures that we use to compare the results produced by the final alignment methods are the following:

- **Precision**, which is the ratio of correctly found correspondences over the total number of correspondences returned by the matching system
- **Recall**, which is the ratio of correctly found correspondences over the total number of all correct correspondences between two ontologies

- **F-Measure**, which is the harmonic mean of Precision and Recall.

F-Measure is the most important measure for the evaluation of the matching process, because it combines the values of Precision and Recall. There is a larger number of correct correspondences when the Recall value is high. Considering the Precision measure, there are less false correspondences when its value is high. Therefore, if the value of F-Measure is high, there is less additional work for the expert to correct obtained correspondences (finding additional correct correspondences and deleting the false found correspondences).

In Figure 3. the performance comparison of the final alignment methods for the entire *Benchmark biblio* ontology track is shown. It can be seen that the final alignment methods IterativeCro (our previous method described in [6]) and IterativeNew (our method presented in this paper) achieved the best matching results (F-Measure = 0.89). The methods Hungarian and Max1 also achieved very good results (F-Measure = 0.87). However, these methods achieved the results worse than our two methods because these methods are not well-balanced i.e. they are adjusted to achieve good results of Precision but not of Recall or vice versa.

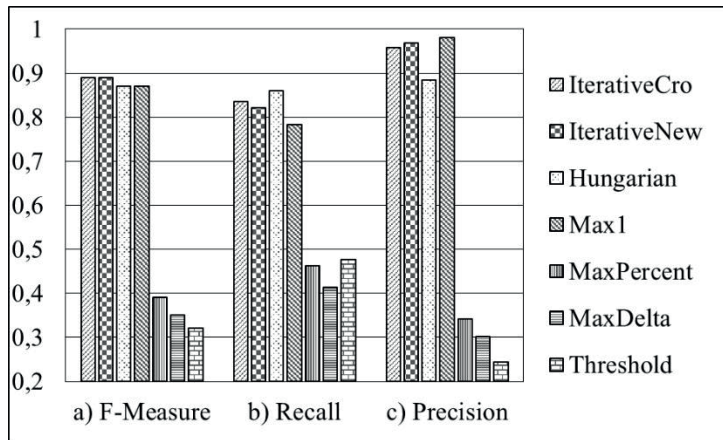


Figure 3. Comparison of the final alignment methods' results for the entire *Benchmark biblio* ontology track

The Max1 method achieved the best results of Precision, but it is the only method, considering the results of the best four methods, for which the result of Recall is less than 0.80. Regarding the Hungarian method, it achieved the best results of Recall, but it is the only method among the best four methods that achieved the result of Precision less than 0.90. Our two methods did not achieved the best results of Precision and Recall but their results are very good for both measures therefore our methods achieved the best overall results that are represented by F-Measure. Considering the methods MaxPercent, MaxDelta and Threshold, these methods achieved very poor results. As stated before, the main problem of these three methods is the possibility of selecting more than one correspondence of the same entity in the final alignment. Since the mapping relationship between entities of different ontologies is usually in

the 1:1 ratio, this possibility of selecting more than one correspondence of the same entity leads to poor results of the matching process.

As stated before, our IterativeNew method achieved the best matching results together with IterativeCro method, but IterativeNew method automatically adjusts the threshold value for determination of the correspondences that will be part of the final alignment which is not the case with our previous method where the threshold is adjusted manually from experience. The manually adjusted threshold has to be defined every time when the system matches a new pair of ontologies. It is difficult for an expert and even more for an ordinary user to adjust this threshold value analyzing the ontologies that have to be matched. Considering the entire ontology matching process, it can never be fully automated, but any increase of the automation within the process is useful if it facilitates the usage of matching system.

Furthermore, analyzing the result of each test case within the *Benchmark biblio* ontology track individually (Table 1.), it can be seen that our two methods produce quite different results for a large number of test cases although their overall results are equal.

Test	IterativeCro			IterativeNew		
	R	P	F-M	R	P	F-M
101	1,000	1,000	1,000	0,980	1,000	0,990
201	1,000	1,000	1,000	0,970	1,000	0,985
201-2	1,000	1,000	1,000	0,918	1,000	0,957
201-4	1,000	1,000	1,000	0,990	1,000	0,995
201-6	1,000	1,000	1,000	0,959	1,000	0,979
201-8	1,000	1,000	1,000	0,990	1,000	0,995
202	0,856	0,933	0,893	0,836	0,988	0,906
202-2	0,980	1,000	0,990	0,980	1,000	0,990
202-4	0,928	0,990	0,958	0,928	1,000	0,963
202-6	0,887	0,978	0,930	0,887	1,000	0,940
202-8	0,877	0,989	0,930	0,866	1,000	0,928
221	1,000	1,000	1,000	0,980	1,000	0,990
222	1,000	1,000	1,000	0,990	1,000	0,995
223	1,000	1,000	1,000	0,970	1,000	0,985
224	1,000	1,000	1,000	0,980	1,000	0,990
225	1,000	1,000	1,000	0,908	1,000	0,952
228	1,000	1,000	1,000	1,000	1,000	1,000
232	1,000	1,000	1,000	0,980	1,000	0,990
233	1,000	1,000	1,000	1,000	1,000	1,000
236	1,000	1,000	1,000	0,970	1,000	0,985
237	1,000	1,000	1,000	1,000	1,000	1,000
238	1,000	1,000	1,000	0,928	1,000	0,963
239	1,000	1,000	1,000	1,000	1,000	1,000
240	1,000	1,000	1,000	1,000	1,000	1,000
241	1,000	1,000	1,000	0,970	1,000	0,985
246	1,000	1,000	1,000	0,970	1,000	0,985
247	1,000	1,000	1,000	0,940	1,000	0,969
248	0,846	0,932	0,887	0,836	0,942	0,886
248-2	0,990	1,000	0,995	0,980	1,000	0,990
248-4	0,928	1,000	0,963	0,918	1,000	0,957

248-6	0,897	1,000	0,946	0,887	1,000	0,940
248-8	0,877	0,989	0,930	0,866	1,000	0,928
249	0,712	0,852	0,776	0,681	0,815	0,742
249-2	0,990	1,000	0,995	0,990	1,000	0,995
249-4	0,949	1,000	0,974	0,959	1,000	0,979
249-6	0,825	0,931	0,875	0,825	0,931	0,875
249-8	0,794	0,906	0,846	0,784	0,895	0,836
250	0,607	1,000	0,755	0,576	0,950	0,717
250-2	0,910	1,000	0,953	0,940	1,000	0,969
250-4	0,819	1,000	0,900	0,849	1,000	0,918
250-6	0,697	0,921	0,793	0,728	0,960	0,828
250-8	0,637	0,955	0,764	0,607	1,000	0,755
251	0,740	0,973	0,841	0,730	0,986	0,839
251-2	0,917	1,000	0,957	0,917	1,000	0,957
251-4	0,813	0,976	0,887	0,813	0,976	0,887
251-6	0,813	0,988	0,892	0,813	1,000	0,897
251-8	0,813	0,988	0,892	0,803	0,988	0,886
252	0,794	0,875	0,833	0,753	0,949	0,840
252-2	0,949	1,000	0,974	0,949	1,000	0,974
252-4	0,897	0,967	0,931	0,887	0,978	0,930
252-6	0,887	0,967	0,925	0,836	0,988	0,906
252-8	0,836	0,953	0,891	0,805	0,963	0,877
253	0,702	0,840	0,765	0,712	0,852	0,776
253-2	0,939	1,000	0,969	0,939	1,000	0,969
253-4	0,897	1,000	0,946	0,897	1,000	0,946
253-6	0,856	0,977	0,913	0,846	1,000	0,917
253-8	0,743	0,858	0,796	0,763	0,903	0,827
254	0,576	0,864	0,691	0,576	0,950	0,717
254-2	0,910	1,000	0,953	0,940	1,000	0,969
254-4	0,849	1,000	0,918	0,879	1,000	0,936
254-6	0,728	1,000	0,843	0,728	1,000	0,843
254-8	0,637	0,914	0,751	0,667	0,917	0,772
257	0,334	0,847	0,479	0,273	0,819	0,410
257-2	0,819	0,932	0,872	0,819	0,932	0,872
257-4	0,788	1,000	0,881	0,788	1,000	0,881
257-6	0,637	0,955	0,764	0,637	0,955	0,764
257-8	0,455	0,883	0,601	0,485	0,889	0,628
258	0,542	0,853	0,663	0,553	0,884	0,680
258-2	0,886	0,978	0,930	0,886	0,978	0,930
258-4	0,792	0,939	0,859	0,813	0,963	0,882
258-6	0,740	0,987	0,846	0,740	0,987	0,846
258-8	0,573	0,874	0,692	0,563	0,932	0,702
259	0,568	0,798	0,664	0,547	0,780	0,643
259-2	0,897	0,967	0,931	0,887	0,967	0,925
259-4	0,825	0,942	0,880	0,815	0,964	0,883
259-6	0,815	0,909	0,859	0,794	0,917	0,851
259-8	0,691	0,828	0,753	0,629	0,848	0,722
260	0,438	0,875	0,584	0,438	0,824	0,572
260-2	0,875	1,000	0,933	0,844	1,000	0,915
260-4	0,844	1,000	0,915	0,844	1,000	0,915
260-6	0,688	1,000	0,815	0,688	0,957	0,801
260-8	0,469	1,000	0,639	0,469	1,000	0,639

261	0,394	0,591	0,473	0,394	0,765	0,520
261-2	0,819	0,900	0,858	0,849	0,934	0,889
261-4	0,667	0,815	0,734	0,728	0,858	0,788
261-6	0,697	0,794	0,742	0,758	0,834	0,794
261-8	0,637	0,875	0,737	0,576	0,950	0,717
262	0,243	0,616	0,349	0,213	0,637	0,319
262-2	0,879	1,000	0,936	0,879	1,000	0,936
262-4	0,697	0,921	0,793	0,697	0,921	0,793
262-6	0,607	1,000	0,755	0,637	1,000	0,778
262-8	0,516	1,000	0,681	0,516	1,000	0,681
265	0,000	1,000	0,000	0,000	1,000	0,000
266	0,122	0,308	0,175	0,091	0,250	0,133

Table 1. The results of each test case within the *Benchmark biblio* ontology track obtained by IterativeNew and IterativeCro methods

To better understand the obtained results, we grouped similar test cases within the *Benchmark biblio* ontology track according to the implemented components within ontologies of these test sets in order to analyze the results which are obtained by IterativeCro and IterativeNew final alignment methods. We made 5 different groups of test cases that consist of ontologies in which: all main components are implemented (ALL), the labels and comments of entities are missing (No-LC), the labels, comments and ontology structure are missing (No-LCS), the labels, comments and instances of entities are missing (No-LCI) and the labels, comments and properties are missing (No-LCP).

In Figure 4. the comparison of the results of Precision measure obtained by IterativeCro and IterativeNew methods for different groups of test cases within *Benchmark biblio* ontology track can be seen.

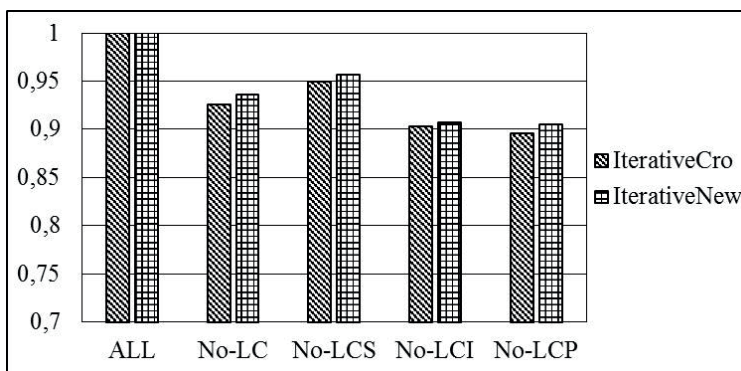


Figure 4. Comparison of the results of Precision measure for different groups of test cases within *Benchmark biblio* ontology track

The IterativeNew method achieved better results of Precision than IterativeCro method for all groups of test cases (No-LC, No-LCS, No-LCI and No-LCP) except for the group ALL where the methods achieved the same results. Obviously, the

IterativeNew method reduced the number of false correspondences (higher Precision value than IterativeCro) within the final alignment.

Considering the Recall measure, the comparison of the results obtained by IterativeCro and IterativeNew methods for different groups of test cases within *Benchmark biblio* ontology track can be seen in Figure 5.

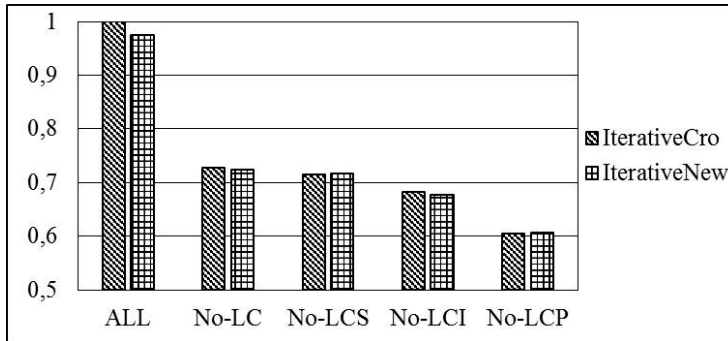


Figure 5. Comparison of the results of Recall measure for different groups of test cases within *Benchmark biblio* ontology track

Here, the IterativeCro method achieved better results of Recall than the IterativeNew method for three groups of test cases (ALL, No-LCS and No-LCI). However, IterativeNew achieved better results of Recall for two groups of test cases (No-LCS and No-LPC). The greatest difference between results of Recall produced by these two final alignment methods is when the test cases consist of ontologies that have all components implemented within them (ALL), especially labels and comments that contain the most important information about entities. The difference between results for the other four groups of test cases is not as significant as for the group ALL. In Figure 6. the comparison of the results of F-Measure for different groups of test cases obtained by IterativeCro and IterativeNew methods can be seen.

Here, IterativeNew method achieved better results for three groups of test cases (No-LC, No-LCI and No-LCP) while IterativeCro method achieved better results for two groups of test cases (ALL and No-LCS). Considering the results presented in this section, IterativeNew method performs better when one or more components within compared ontologies are missing. There are more false correspondences when all components within ontologies are not implemented, therefore IterativeNew method recognizes these correspondences better than IterativeCro according to the Precision value. We can assume that the introduction of new restriction rule (ThresholdSecond) together with the automatic adjustment of threshold value described in Section 4 reduced the number of false correspondences within the final alignment and consequently increased the Precision measure. However, these elements lead to lower Recall value because of their additional regulations while selecting correspondences for the final alignment. Therefore, considering all results together, we can assume that there is still room for improvement in the IterativeNew final alignment method that will lead to better results of Recall while preserving very high Precision.

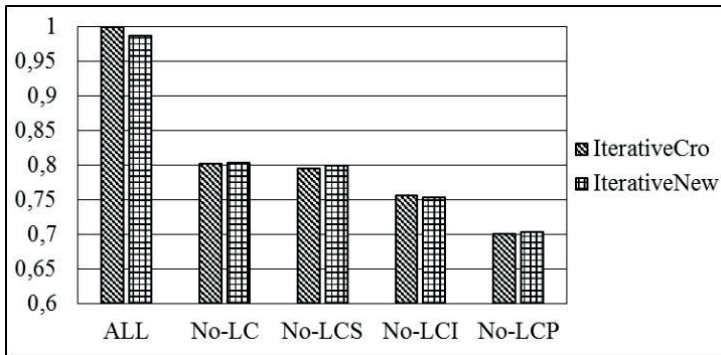


Figure 6. Comparison of the results of F-Measure for different groups of test cases within *Benchmark* biblio ontology track

Comparing the final alignment process of IterativeNew and IterativeCro methods, it is important to notice that our IterativeNew method adjusts the threshold in the final alignment process automatically as opposed to IterativeCro method where the threshold value is adjusted manually based on user's experience. As stated before, any increase of automation within the ontology matching process facilitates the usage of matching system for non-expert users.

6. Conclusion

In this paper we proposed a new version of our iterative final alignment method for automatically determining the correspondences between compared ontologies that enter the final alignment. The final alignment method is one of the most important components within the ontology matching system because it is the last subprocess within the matching process. As stated before, if the final alignment method does not select correct correspondences in the final alignment, then the system will not perform the quality matching process for any pair of compared ontologies.

We improved our final alignment method by introducing the automatic adjustment of the threshold value for selecting the proper correspondences in the final alignment. Automatic adjustment of threshold value facilitates the use of entire ontology matching system. Furthermore, we proposed a new restriction rule of discarding a certain correspondence with highest value between two entities from the final alignment if there is no significant difference between the values of the highest and the second highest correspondence of one of the compared entities. Thereby, two correspondences, that have approximately the same values and contain the same entity of an ontology, are excluded from the final alignment. This restriction rule decreases the number of false correspondences within the final alignment.

The evaluation of final alignment methods were performed on *Benchmark* biblio ontology track that contains more than 100 pairs of ontologies and the alignments results between them. All tested final alignment methods were integrated within our CroMatcher ontology matching system in order to perform the evaluation of these methods in the same environment. The evaluation results have shown that our new

version of iterative final alignment method achieved the best overall results together with our previous final alignment method. However, our new method works completely automatically as opposed to our previous method and it is the great advantage that facilitates the usage of the matching system.

Furthermore, by analyzing the results of a groups of similar test cases within the *Benchmark* biblio track, it can be concluded that our new method achieved the best results when one or more components within compared ontologies are missing. Furthermore, implementation of our new restriction rule resulted with the smallest number of false correspondences found within the alignment compared to all approaches in the evaluation. In our future work particular attention will be given to the increase of the number of correct correspondences found within the final alignment, while preserving the obtained small number of false correspondences found within the alignment.

References

- [1] P. Borst, H. Akkermans, and J. Top, "Engineering ontologies," *Int. J. Hum. Comput. Stud.*, vol. 46, no. 2–3, pp. 365–406, 1997.
- [2] G. Antoniou and F. van Harmelen, *Semantic Web Primer*. MIT Press, 2004.
- [3] T. Berners-Lee and C. J. Jaffe, "World Wide Web Consortium (W3C)," *Web Inventors*, 2013. [Online]. Available: <http://www.w3.org/>
- [4] J. Euzenat and P. Shvaiko, *Ontology matching*, 2nd Edition. Springer-Verlag, Berlin Heidelberg, 2013.
- [5] A. Broder, "A Taxonomy of Web Search," *SIGIR Forum*, vol. 36, no. 2, pp. 3–10, 2002.
- [6] M. Gulić, B. Vrdoljak, and M. Banek, "CroMatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment," *J. Web Semant.*, vol. 41, pp. 50–71, 2016.
- [7] J. Euzenat, C. Meilicke, H. Stuckenschmidt, P. Shvaiko, and C. Trojahn, "Ontology Alignment Evaluation Initiative : Six Years of Experience," *J. Data Semant. XV, LNCS 6720*, vol. 6720, pp. 158–192, 2011.
- [8] E. Jimenez-Ruiz, "Ontology alignment evaluation initiative," 2016. [Online]. Available: <http://oaei.ontologymatching.org/>
- [9] H. Do and E. Rahm, "COMA: a system for flexible combination of schema matching approaches," in *Proceedings of the 28th international conference on Very Large Data Bases, Hong Kong SAR, China, 2002*, pp. 610–621.
- [10] D. Aumuellner, H. Do, S. Massmann, and E. Rahm, "COMA++ - Schema and ontology matching with COMA," In *Proceedings of the 24th*

- International Conference on Management of Data (SIGMOD), Baltimore, Maryland, USA, 2005, pp. 906–908.
- [11] S. Hertling, "Hertuda Results for OEAI 2012," in *CEUR Workshop Proceedings*, 2012, vol. 946, pp. 141–144.
- [12] T.T. Dang, A. Gabriel, S. Hertling, P. Roskosch, M. Wlotzka, J.R. Zilke, F. Janssen and H. Paulheim, "HotMatch results for OEAI 2012," in *CEUR Workshop Proceedings*, 2012, vol. 946, pp. 145–151.
- [13] Y. Zhang, W. Wang, S. He, K. Liu, J. Zhao and X. Lv, "IAMA results for OEAI 2013," in *CEUR Workshop Proceedings*, 2013, vol. 1111, pp. 123–130.
- [14] I.H. Kuo and T.T. Wu, "ODGOMS - results for OEAI 2013," in *CEUR Workshop Proceedings*, 2013, vol. 1111, pp. 153–160.
- [15] M. Ehrig and Y. Sure, "Ontology mapping – an integrated approach," *Lecture Notes in Computer Science: The Semantic Web: Research and Applications*, vol. 3053, pp. 76–91, May 2004.
- [16] J. Tang, J. Li, B. Liang, H. Xiaotong, Y. Li, and K. Wang, "Using Bayesian decision for ontology mapping," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 4, no. 4, pp. 243–262, 2006.
- [17] T. Bach, R. Dieng-Kuntz and F. Gandon, "On ontology matching problems (for building a corporate semantic web in a multi-communities organization)," in *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS)*, Porto, Portugal, 2004, pp. 236–243.
- [18] J. Garcia and K. Asooja, "Monolingual and cross-lingual ontology matching with CIDER-CL: evaluation report for OEAI 2013," in *CEUR Workshop Proceedings*, 2013, vol. 1111, pp. 109–116.
- [19] N. Jian, W. Hu, G. Cheng, and Y. Qu, "Falcon-AO: Aligning ontologies with falcon," in *CEUR Workshop Proceedings*, 2005, vol. 156, pp. 85–91.
- [20] Y. Qu, W. Hu and G. Cheng, "Constructing virtual documents for ontology matching," in *Proceedings of the 15th International World Wide Web Conference*, Edinburgh, Scotland, UK, 2006, pp. 23–31.
- [21] D. Ngo and Z. Bellahsene, "YAM++ - Results for OEAI 2013," in *CEUR Workshop Proceedings*, 2013, vol. 1111, pp. 211–218.
- [22] D. Ngo and Z. Bellahsene, "YAM++ - Results for OEAI 2012," in *CEUR Workshop Proceedings*, 2012, vol. 946, pp. 226–233.
- [23] D. Ngo, Z. Bellahsene and R. Coletta, "YAM++ results for OEAI 2011," in *CEUR Workshop Proceedings*, 2011, vol. 814, pp. 228–235.

- [24] D. Ngo, Z. Bellahsene and R. Coletta, "A generic approach for combining linguistic and context profile metrics in ontology matching," *Lecture Notes in Computer Science: On the Move to Meaningful Internet Systems*, vol. 7045, pp. 800–807, October 2011.
- [25] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, no. 1-2, pp. 83-97, March 1955.
- [26] J. Euzenat, M. E. Roşoiu, and C. Trojahn, "Ontology matching benchmarks: Generation, stability, and discriminability," *J. Web Semant.*, vol. 21, pp. 30–48, 2013.
- [27] J. Euzenat, "OAEI 2016, results for Benchmark biblio ontology track," 2016. [Online]. Available: <http://oaei.ontologymatching.org/2016/results/benchmarks/index.html>