

# Supervised Machine Learning Models and Schema Matching Techniques for Ontology Alignment

Faten Abbassi<sup>a</sup> and Yousra Bendaly Hlaoui<sup>b</sup>

LIPSIC Laboratory, University of Tunis El Manar, Faculty of Sciences of Tunis El Manar, Tunisia

**Keywords:** Ontology Alignment, Machine Learning, Schema Matching, Reference Ontologies, Conference Track, Benchmark Track.


**Abstract:** The diversity of existing representations of the same ontology creates a problem of manipulation of the same knowledge according to any computational domain. Unifying similar ontologies by reducing their degree of heterogeneity seems to be the appropriate solution to this problem. This solution consists of aligning similar ontologies using a set of existing ontology schema-matching techniques. In this paper, we present an approach for ontology alignment based on these techniques and machine learning models. To do so, we have developed a matrix construction method based on ontology matching techniques, namely element matching techniques and structure matching techniques implemented by elementary matchers. Once the matrix is constructed, we apply a composite matcher, which is a classifier to combine the individual degrees of similarity calculated for each pair of ontology elements into a final aggregated similarity value between the two ontologies. This composite matcher is implemented via various supervised machine learning models such as *LogisticRegression*, *GradientBoostingClassifier*, *GaussianNB* and *KNeighborsClassifier*. To experiment our alignment method and to validate the used learning models, we used the reference ontologies and their alignments for the *conference* and *benchmark* tracks provided by the *Ontology Alignment Evaluation Initiative* (OAEI <sup>a</sup>).


<sup>a</sup><http://oaei.ontologymatching.org/>

## 1 INTRODUCTION

An ontology, as defined by Gruber (Gruber and Olsen, 1994), is a formal and explicit specification of a shared conceptualisation. It is used to conceptualise knowledge using concepts or classes, relationships between these classes and individuals instantiating these classes. However, the diversity of representations of the same ontology can lead to difficulties in knowledge management and manipulation. To remedy this problem, we propose to unify similar ontologies by reducing their heterogeneity through a process of ontology alignment. This process aims to compute similarity measures between the different entities or elements of each pair of ontology schema (Euzenat et al., 2007) based on existing schema matching techniques (Shvaiko and Euzenat, 2005). Schema matching techniques (Rahm and Bernstein, 2001; Euzenat et al., 2007) are based on two main aspects: (i) the granularity of matching, i.e. at the level of elements

(ontology classes or individuals) or structure (relationships between classes) and (ii) the way these techniques interpret input information (class labels, data properties and relationships). These techniques are implemented using individual matchers (Rahm and Bernstein, 2001), which calculate similarity according to input interpretation criteria corresponding to each level of ontology granularity. There are two types of interpretation: syntactic and external. When the input is interpreted according to the *syntactic criterion*, it is considered as a sequence of characters specified by a defined syntactic structure. On the other hand, when the input is interpreted according to the *external criterion*, it is seen as a linguistic object, using external resources such as a thesaurus to express relationships between the terms of the labels. Thereafter, the *composite matchers* combine the results of the different individual matchers, which were used independently according to different criteria, to provide a final decision on ontology similarity. Hence, we have implemented the composite matchers using supervised machine learning models, as we used la-

<sup>a</sup>  <https://orcid.org/0000-0001-7525-4505>

<sup>b</sup>  <https://orcid.org/0000-0002-3476-0185>

beled data (labels of ontological entities) and continuous values (values of ontological similarity measures in the interval  $[0, 1]$ ). Several studies (Bulygin, 2018; Bulygin and Stupnikov, 2019; Xue and Huang, 2023) have explored the application of machine learning to ontology matching and they have shown that this approach can improve the accuracy and efficiency of ontology alignment. However they presented a number of limitations, such as:

1. No respect of the ontology structure in the alignment process (Xue and Huang, 2023).
2. Incorrect values of similarity measures for similar ontology classes.
3. Reduced number of matching techniques are used to compute ontology element similarity measures.

To overcome these limitations and achieve a high level of accuracy by the ontology alignment process, we propose, in this paper, an ontology alignment approach based on supervised machine learning models and various schema matching techniques with respect to the ontology structure.

The remainder of this paper is organised as follows. Section 2 presents a discussion of related work. Section 3 presents our alignment approach. Section 4 presents the Experimentation and validation. This article is discussed in section 5. We conclude this paper in section 6 and propose some perspectives.

## 2 RELATED WORK

In the literature, several works have been performed to align ontologies. Some of them are based on machine learning, such as those published in (Bulygin, 2018; Bulygin and Stupnikov, 2019; Xue and Huang, 2023).

The work proposed in (Bulygin, 2018) exploits lexical and semantic information as inputs to the machine learning models *NaiveBayesClassifier*, *LogisticRegression* and *XGBoost*. Unfortunately, this work delivers identical entities with incorrect similarity measures and provides very low precision and accuracy rates compared to the work proposed in (Bulygin and Stupnikov, 2019). Authors in (Bulygin and Stupnikov, 2019) have proposed an approach that combines 29 similarity techniques based on strings, languages, and structures to build their data matrices. They have used *LogisticRegression*, *RandomForestClassifier*, and *GradientBoosting* as machine learning models. While authors in (Xue and Huang, 2023) have proposed an ontology alignment approach based on the unsupervised machine learning method of the generative adversarial network with a simulated

annealing algorithm (SA-GAN). The used similarity measure techniques include string-based techniques such as Levenshtein distance, Jaro distance, Dice coefficient, N-gram, and the WordNet language-based technique. The principal limit of the approaches proposed in (Bulygin and Stupnikov, 2019; Xue and Huang, 2023) is to consider entities in the alignment process independently from their data properties and object properties, which affects the alignment accuracy.

Based on this comparative study, the originality of our contribution is defined as follows:

- The alignment of ontological classes is according to their data properties and their relationship or object properties. However, the approaches presented in (Bulygin, 2018; Bulygin and Stupnikov, 2019) align classes independently of their data properties and their relationship properties which decreases the accuracy.
- The alignment process respects the ontology's structure, whereas this aspect is not taken into account in the approach proposed by (Xue and Huang, 2023).

## 3 PROPOSED APPROACH

Our approach consists in matching ontologies based on the use of different machine learning models to combine the individual degrees of similarity calculated for each pair of ontology elements in a final aggregated similarity value. Indeed, the accuracy and efficiency of our approach are based on the similarity measure matrices that we have constructed, which are the input to the used machine learning models. According to figure 1, this approach is mainly composed of three principal phases: the *Pre-Processing* phase, the *Training and Testing* phase and the *Quality Evaluation* phase (cf. Figure 1).

### 3.1 Phase 1: Pre-Processing

As shown in figure 1, this phase takes as input a pair of *reference ontologies* and their corresponding *reference alignment* files provided by the OAEI (Ondřej Zamazal, ). The output of this phase is a matrix containing the calculated similarity values and the reference alignment values. This matrix is constructed in two steps: *Ontology Element Extraction* step and *Similarity Value Calculation* step.

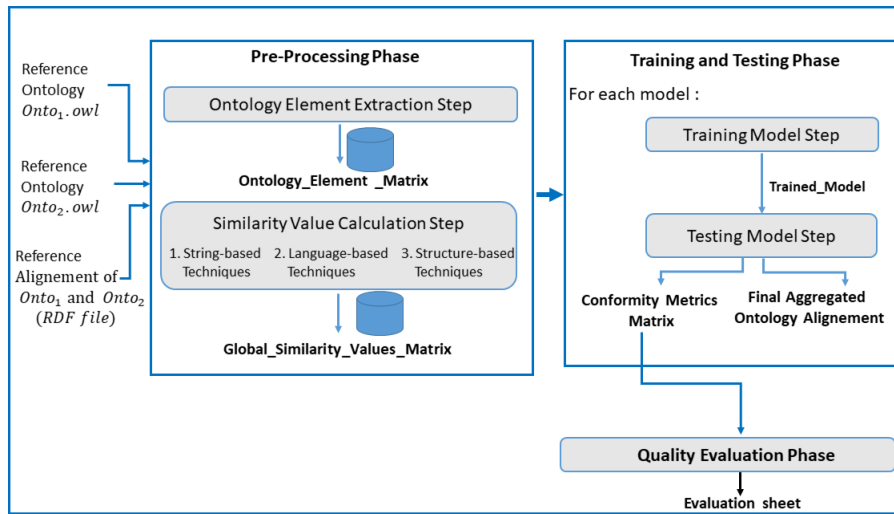


Figure 1: Architecture of the proposed approach.

### 3.1.1 Step 1.1: Ontology Element Extraction

This step consists of extracting the ontological elements needed to build the ontological element matrix from the OWL files of the ontologies of the input reference ontology pairs, provided by the OAEI competition, as well as the confidence values, calculated by the OAEI competition, from their input reference alignment files (RDF files). The result is a matrix of ontological elements, including *class labels*, *data property labels*, *relationship labels* or *objectProperty labels* as indicated in the OWL file. In addition, the confidence value (calculated by OAEI) is extracted from the reference alignment file.

The process of constructing the ontological element matrix is detailed as follows:

- **Step 1:** extracting the classes lists, respectively named *Classes1* and *Classes2*, of the two ontologies *Onto1* and *Onto2*.
- **Step 2:** for each class  $CL_i$  of the *Classes1* list, construct a pair of classes in the form  $(CL_i, CL_j)$  with all the classes  $CL_j$  of the *Classes2* list with  $i \geq 1$  and  $j \geq 1$ . More precisely, applying the Cartesian product of all classes  $CL_i$  from list *Classes1* with all classes  $CL_j$  from list *Classes2*.
- **Step 3:** for each pair of constructed classes  $(CL_i, CL_j)$ , we extract from the OWL files of the two ontologies *Onto1* and *Onto2* the list of data properties of each of the two classes  $CL_i$  and  $CL_j$ , named respectively *Data\_Properties\_CL<sub>i</sub>* and *Data\_Properties\_CL<sub>j</sub>*, as well as the list of object properties of each of the two classes  $CL_i$  and  $CL_j$ , named respectively *relationships\_CL<sub>i</sub>* and *relationships\_CL<sub>j</sub>*. In addition, we extract the confidence value, named *Confident\_alignment*, of the

two classes  $CL_i$  and  $CL_j$  from the reference alignment file (RDF file).

- **Step 4:** for each pair of classes  $(CL_i, CL_j)$ , we construct a vector called *VElements*, containing all the data extracted by the end of step 3. This vector is defined by:

$$VElements = (Onto1, Onto2, CL_i, CL_j, Dataproperties\_CL_i, Dataproperties\_CL_j, relationships\_CL_i, relationships\_CL_j, Confident\_alignment)$$

- **Step 5:** we construct the *Ontology\_Element\_Matrix*, where each of its row is an instance of the *VElements* vector. The size of the *Ontology\_Element\_Matrix* is  $n * m$  where  $n$  and  $m$  are respectively the number of classes of the ontologies *Onto1* and *Onto2* to align.

### 3.1.2 Step 1.2: Similarity Value Calculation

This step consists of computing the syntactic and external similarity measures for the different pairs of entities stored in the *Ontology\_Element\_Matrix*. The output is a similarity matrix containing the computed similarity values and the reference alignment values of the ontology elements stored in the input matrix. To do this, we have used 24 individual matchers implementing 21 *string-based* (Bulygin and Stupnikov, 2019) techniques and 3 *language-based* (Bulygin and Stupnikov, 2019) techniques (cf. Table1).

Thus, the computation of the similarity measure is detailed as follows :

- **Step 1:** we Apply a normalisation to the ontological elements stored in the input matrix, i.e. *class labels*, *data property labels* and *relation-*

Table 1: String, language and structure-based techniques used by our approach.

Technique class	Techniques
String-based techniques	N-gram 1, N-gram 2, N-gram 3, Dice coefficient, Jaro measure, Monge-Elkan, Smith-Waterman, Needleman-Wunsh, Affine gap, Bag distance, Cosine similarity, Partial Ratio, Soft TF-IDF, Generalized Jaccard, Jaro-Winkler, Partial Token Sort Fuzzy Wuzzy Ratio, Soundex, TF-IDF, Token Sort, TverskyIndex, Overlap coefficient, and Longest common subsequence (Euzenat et al., 2007; Bulygin, 2018).
Language-based techniques	Wu and Palmer similarity, Word2vec and Spacy (Euzenat et al., 2007).
Structure-based techniques	Apply all string-based and language-based techniques between two class labels, data property labels and relationship labels of two ontological entities.

ship labels. Our objective is to transform these entities to a common format in order to enhance the alignment result. We have used the normalisation techniques: *case normalisation*, *blank normalisation*, *link striping*, *punctuation elimination*, *diacritics suppression* and *digit suppression*.

- **Step 2:** we apply different individual matchers to compute the similarity measures for each pair of normalised ontological elements.
- **Step 3:** we construct a vector for each pair of ontological elements containing the calculated similarity measures. We distinguish *VSim\_Classes* vector, *VSim\_Properties* vector and *VSim\_relationships* vector. Each of these vectors is defined on 24 similarity values and takes the following form:

$$VSim\_Entity = (\text{sim\_Ngram}, \text{sim\_Wordnet}, \dots, \text{sim\_jaro}, \text{sim\_Spacy})$$

Where Entity denotes classes, data properties or relationships.

- **Step 4:** we combine the constructed vectors into a global similarity vector called *V\_GSim*, which contains all the calculated values, as well as the confident alignment of each pair of elements concerned. The *V\_GSim* vector is defined as follows:

$$V\_GSim = (VSim\_Classes, VSim\_Properties, VSim\_relationships, Confident\_alignment)$$

This vector is constructed for each pair of classes of the ontologies to be aligned. Thus, this vector contains 72 similarity values (3 VSIM vectors \* 24 techniques) and the confident alignment of the current pair of classes.

- **Step 5:** we build the similarity matrix *Global\_Similarity\_Values\_Matrix*, where each of its row represents an instance of the computed *V\_GSim*. The *Global\_Similarity\_Values\_Matrix* has the same size as the *Ontology\_Element\_Matrix* created by the previous step.

### 3.2 Phase 2: Training and Testing

This phase consists in determining the final aggregated alignment of a pair of ontologies. It takes as input the *Global\_Similarity\_Values\_Matrix* provided by the previous phase and four machine learning models. As output, the training and testing phase provide the degree of similarity of a pair of input ontologies, as well as the efficiency measures including precision, recall and f-measure values, provided by each used machine learning model. We have used the *LogisticRegression*, *GradientBoostingClassifier*, *GaussianNB* and *KNeighborsClassifier* models that are most frequently used in the literature (Bulygin, 2018; Bulygin and Stupnikov, 2019). We trained each of these models using a training matrix built of 60% of the number of rows of the *Global\_Similarity\_Values\_Matrix*. Then we tested each of these trained models using a test matrix built of the remaining 40% of the number of rows in the *Global\_Similarity\_Values\_Matrix*. For each pair of ontologies, each machine learning model used builds the necessary datasets to evaluate the degree of similarity between the ontologies, based on the first 72 columns of the training matrix. This creates a trained model. Then, this model uses the first 72 columns of the corresponding test matrix to provide a final classification of the current ontology pair, evaluating their similarities according to the confident alignment value (column *Confident\_alignment*). This classification is based on the conformity measures that evaluate the degree of correspondence between the degrees of similarity predicted by each machine learning model and the confident value of alignment. Indeed, we have used the *accuracy* or *precision* (P), *recall* (R) and *f-measure* (Euzenat et al., 2007) metrics as conformity metrics. These metrics are the most frequently used in this context. They are defined as follows:

$$P : \Lambda \times \Lambda \rightarrow [0..1]$$

$$P(A, T) = \frac{|T \cap A|}{|A|}$$

$$R : \Lambda \times \Lambda \rightarrow [0..1]$$



$$R(A, T) = \frac{|T \cap A|}{|T|}$$

$$f - measure = \frac{2 * P(A, T) * R(A, T)}{P(A, T) + R(A, T)}$$

Where  $A$  is the set of all values of calculated alignments and of reference alignments provided by **OAEI**,  $T$  is the set of all values of reference alignments,  $A$  is the set of all values of calculated alignments and  $|A \cap T|$  is the cardinality of the set of values of calculated alignments according to the values of the reference alignments.

### 3.3 Stage 3: Quality Evaluation

This phase takes as input the matrices of conformity metrics, provided by the previous phase, for the reference ontology pairs that we have used in our approach. It consists of comparing the results provided by our approach with those of various OAEI approaches, in particular (Bulygin and Stupnikov, 2019; Huber et al., 2011; Bock et al., 2011; David, 2011; Xue and Huang, 2023; David, 2007; Eckert et al., 2009; Straccia and Troncy, 2005; Euzenat et al., 2005). Notably, the ultimate goal of this comparison is the validation of the machine learning models. It is based mainly on the table of f-measure values returned by each model for all the alignment tests that we have performed on the reference ontology pairs (see Table 3 and Table 4). We chose to use this metric in this comparison because it calculates the harmonic mean of *precision* and *recall*, giving them the same importance (Euzenat et al., 2007) (see section 3.2).

## 4 EXPERIMENTATION AND QUALITY EVALUATION OF THE ALIGNMENT APPROACH

To implement different phases of our approach, we have used *Python* with *anaconda1.10.1* and the *Spyder5.0.3* editor, which are configured by a set of tools, namely the dictionary *GoogleNews-vectors-negative3-00* and the libraries *py.stringmatching*, *beautifulsoup4*, *Owlready2*, *pandas*, *fuzzycomp*, *NGram*, *Wordnet*, *nlk* (Natural Language Toolkit), *spacy*, *en\_core\_web\_lg*, *textbfGensim*, *tqdm*, *Keras* and *sklearn*. These tools are executed on a laptop with a 64-bit operating system, an X64 Intel Core processor *i7-8550U 1.80GHz - 1.99GHz* with a version of Windows 10 Professional *N* and a RAM of 8.00G bytes.

### 4.1 Hyper Parameter Tuning of Used Machine Learning Models

Table 2 summarises the hyper parameter tuning for *LogisticRegression*, *GradientBoostingClassifier* and *KNeighborsClassifier* models, as the *GaussianNB* model does not need to be configured.

Table 2: hyper parameter tuning of used machine learning models.

Model	hyperparameter
LogisticRegression	max_iter= 1000, solver='lbfgs'
GradientBoosting Classifier	learning_rate = 1, n_estimators= 100
KNeighborsClassifier	n_neighbors=1

### 4.2 Used Reference Ontology Tracks

In our approach, we are focused on the *benchmark* track and the *conference* track among the various tracks provided by the OAEI (Ondřej Zamazal, ) competition. Each track consists of a set of reference ontologies (OWL<sup>1</sup> files) and their reference alignments (RDF<sup>2</sup> files). Indeed, the *benchmark* track consists of a collection of reference ontologies from various domains and of different sizes. This collection includes the reference ontology, Ontology 101 (OWL file), as well as several variations of this ontology (Ondřej Zamazal, ). These variations (OWL files) are systematically generated from Ontology 101 by deleting certain ontological information in order to evaluate the performance of our algorithms in the absence of this information. The ontology variations are classified into three test families. The first, the simple **1xx** test family, compares the reference ontology to itself, to an irrelevant ontology, or to an ontology with linguistic restrictions and language generalisation. The second, systematic test family **2xx**, involves deleting or replacing ontology components with synonyms, random strings or strings in another language. Finally, the third, the **3xx** family, consists of four real ontologies reminiscent of BibTeX namely the ontologies 301, 302, 303 and 304. In our contribution, we have used ontologies 101 and 104 from family **1xx**, ontologies 201, 208, 221, 247, 248 and 266 from family **2xx** and all ontologies from family **3xx**. The *conference* track shows the highest degree of heterogeneity compared to the other tracks. This is an essential feature for the ontology alignment task. It consists of seven reference ontologies (OWL files).

<sup>1</sup><https://www.w3.org/OWL/>

<sup>2</sup><https://www.w3.org/RDF/>

Table 3: Values of the f-measure of our approach compared to the approach proposed by (Bulygin and Stupnikov, 2019) for each pair of tests in the 2023 OAEI conference track.

Pair of Reference Ontologies	Our Approach				Approach in (Bulygin and Stupnikov, 2019)		
	LR	GBC	GNB	KN	LR	RF	XGB
cmt-conference	0.30	0.30	0.45	0.44	-	-	-
cmt-confOf	0.45	0.44	0.50	0.51	0.44	0.41	0.48
cmt-edas	0.92	0.91	0.75	0.79	0.72	0.76	0.63
cmt-ekaw	0.66	0.60	0.69	0.72	0.58	0.62	0.70
cmt-iasted	0.80	0.79	0.85	0.82	0.88	0.88	0.88
cmt-sigkdd	0.79	0.75	0.70	0.81	0.73	0.80	0.73
conference-confOf	0.69	0.60	0.61	0.75	0.61	0.54	0.57
conference-edas	0.55	0.55	0.60	0.59	0.53	0.5	0.55
conference-ekaw	0.38	0.35	0.51	0.45	0.43	0.40	0.47
conference-iasted	0.45	0.42	0.70	0.66	-	-	-
conference-sigkdd	0.66	0.62	0.65	0.63	0.64	0.54	0.58
confOf-edas	0.55	0.59	0.70	0.55	0.62	0.62	0.62
confOf-ekaw	0.45	0.50	0.60	0.43	0.58	0.68	0.64
confOf-iasted	0.55	0.55	0.55	0.42	0.71	0.61	0.66
confOf-sigkdd	0.79	0.78	0.79	0.76	0.72	0.72	0.72
edas-ekaw	0.56	0.52	0.96	0.90	-	-	-
edas-iasted	0.38	0.45	0.75	0.32	0.42	0.57	0.57
edas-sigkdd	0.70	0.72	0.75	0.66	0.53	0.63	0.63
ekaw-iasted	0.62	0.70	0.55	0.44	0.58	0.75	0.70
ekaw-sigkdd	0.70	0.70	0.77	0.79	0.77	0.77	0.77
iasted-sigkdd	0.90	0.80	0.80	0.75	0.75	0.81	0.81

LR: *LogisticRegression*, GBC: *GradientBoostingClassifier*, GNB: *GaussianNB*, KN: *KNeighborsClassifier*, RF: *RandomForest*, XGB: *XGBoost*.

Table 4: Comparison of our approach with the results obtained by participants in the 2016 OAEI benchmark test in terms of f-measure.

Pair of Reference Ontologies	Approaches in the Literature							Our Approach			
	Falcon	GeRMeSMB	CODI	MapPSO	AROMA	edna	GAN	GBC	GNB	KN	LR
101-104	1.00	1.00	0.99	1.00	0.98	1.00	1.00	0.95	0.55	0.75	0.80
201-208	0.84	0.88	0.45	0.69	0.73	0.54	0.79	0.66	0.45	0.65	0.77
221-247	0.99	0.97	0.98	0.98	0.95	0.88	0.99	0.91	0.88	0.92	0.90
248-266	0.50	0.60	0.37	0.48	0.37	0.35	0.55	0.44	0.55	0.42	0.38
301-304	0.79	0.47	0.59	0.34	0.62	0.46	0.78	0.60	0.75	0.66	0.79

Pair of Reference Ontologies	FOAM	XGBoost	OLA	OMAP	LR	RF	DT	GBC	GNB	KN	LR
	101-302	0.77	0.72	0.34	0.74	0.72	0.71	0.75	0.77	0.70	0.75
101-303	0.84	0.75	0.44	0.84	0.82	0.82	0.81	0.75	0.80	0.88	0.90
101-304	0.95	0.91	0.69	0.91	0.90	0.91	0.96	0.97	0.91	0.90	0.88

LR: *LogisticRegression* of (Bulygin and Stupnikov, 2019), GBC: *GradientBoostingClassifier*, GNB: *GaussianNB*, KN: *KNeighborsClassifier*, XGBoost: *XGBoost* from (Bulygin and Stupnikov, 2019) NN: *Neural Network*, DT: DT of (Eckert et al., 2009) GAN: Generative Adversarial Network from (Xue and Huang, 2023).

To evaluate the performance of ontology matching processes, it is necessary to use reference alignments. These alignments are available as a set of RDF files on the OAEI competition website. Each file contains only similar entities in ontology pairs, and their confidence

value, generally equal to 1.0. We have selected eight reference alignment cases from the *benchmark* track, as they are the most frequently used by researchers in the literature. In addition, we have used 21 reference alignment cases of the *conference* track.

### 4.3 Quality Evaluation of the Alignment Approach

To validate machine learning models that we have used in our approach, we have developed the evaluation process shown in figure 2. This process is based on the comparison of f-measure values provided by each of used machine learning models (cf. Table 3 and Table 4). To evaluate our alignment approach, we have used the reference ontologies and their alignments namely the *conference* and *benchmark* tracks provided by the OAEI. These datasets are frequently used by various approaches in the literature (Bulygin and Stupnikov, 2019; Huber et al., 2011; Bock et al., 2011; David, 2011; Xue and Huang, 2023; David, 2007; Eckert et al., 2009; Straccia and Troncy, 2005; Euzenat et al., 2005).

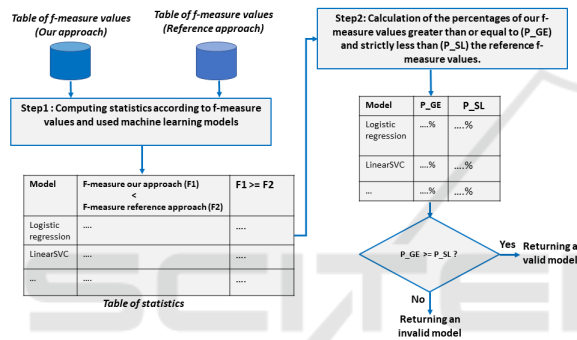


Figure 2: Validation process.

After running the evaluation process (cf. Figure 2) on the ontologies of the conference track, the comparison of our results and those published in (Bulygin and Stupnikov, 2019) leads to the following observation:

1. For the *LogisticRegression* model, **61.60%** of our alignment tests are better than the tests of the alignment approach proposed in (Bulygin and Stupnikov, 2019).
2. For the *GradientBoostingClassifier* model, **57.14%** of our alignment tests are better than the tests of the alignment approach proposed in (Bulygin and Stupnikov, 2019).
3. For the *GaussianNB* model, **80.95%** of the alignment tests provided by our approach are better than those provided by the alignment approach proposed in (Bulygin and Stupnikov, 2019).
4. For the *KNeighborsClassifier* model, **57.14%** of our alignment tests are better than those performed by the alignment approach published in (Bulygin and Stupnikov, 2019).

After executing the evaluation process of figure 2 on

the ontologies of the *benchmark* track, the results of the comparison indicate that :

1. For the *GradientBoostingClassifier* model and the *KNeighborsClassifier* model, **50%** of the alignment tests provided by our approach are better than the tests applied by (Bulygin and Stupnikov, 2019; Huber et al., 2011; Bock et al., 2011; David, 2011; Xue and Huang, 2023; David, 2007; Eckert et al., 2009; Straccia and Troncy, 2005; Euzenat et al., 2005).
2. For the *GaussianNB* model and the *LogisticRegression* model, **75%** of our alignment tests are better than the tests of the approaches proposed by (Bulygin and Stupnikov, 2019; Huber et al., 2011; Bock et al., 2011; David, 2011; Xue and Huang, 2023; David, 2007; Eckert et al., 2009; Straccia and Troncy, 2005; Euzenat et al., 2005).

Hence, the validation process shows that all of models that we have used in our alignment approach are valid.

## 5 DISCUSSION

We have evaluated our approach according to the degree of accuracy of the ontology alignment generated by our approach compared to the alignment accuracy generated by the existing approaches published in (Bulygin and Stupnikov, 2019; Huber et al., 2011; Bock et al., 2011; David, 2011; Xue and Huang, 2023; David, 2007; Eckert et al., 2009; Straccia and Troncy, 2005; Euzenat et al., 2005). It is clear that our approach produces good results on the majority of alignment tests that we have performed on the *conference* and *benchmark* tracks provided by OAEI. This is because our approach exploits ontological data more effectively than the existing approaches. In particular, the ontology scanning process that we have used respects rigorously the structure of the ontologies to be aligned. In addition, the machine learning models that we have used, are more efficient than those employed in the existing approaches.

However, our approach shows low accuracy for some reference ontology pairs such as *confof-iaisted*, *confof-ikaw*, *edas-ekaw*, *cmt-conference*, *cmt-confOff*, *conference-ekaw*, *conference-iaisted*, *201-208*, and *248-266*. This decrease in accuracy is mainly due to the absence of the data type properties (*dataProperty*) and object properties (*objectProperty*) in the structure of each of these ontologies. These properties play a crucial role to increase the accuracy of our alignment approach. Consequently, the absence of these elements in a given ontology schema considerably reduces the accuracy of the alignment.

Therefore, our ontology alignment approach remains a prospective when we use the maximum of the ontology schema elements. In fact, we will consider other ontology elements such as sub-classes, individuals, to enhance our approach's accuracy.

## 6 CONCLUSION AND PERSPECTIVES

In this paper, we have proposed an ontology alignment approach based on several schema matching techniques and machine learning models. We have detailed the different phases and steps that compose this alignment approach, namely the **Pre-Processing** phase, the **Ontology Element Extraction** step, the **Similarity Value Calculation** step, the **Training and Testing** phase and the **Quality Evaluation** phase. The *pre-processing* phase involves building similarity matrices using individual matching tools executed on the reference ontologies provided by the OAEI competition, in particular the *Conference* track and the *Benchmark* track. The *training and testing* phase consists of determining the final aggregated alignment of a pair of ontologies. The *Quality Evaluation* phase consists of comparing the results obtained by our approach with those of various OAEI participants, in order to validate or invalidate the used machine learning models. We have validated our approach by performing experimental results, which give better accuracy than the approaches described in (Bulygin and Stupnikov, 2019; Huber et al., 2011; Bock et al., 2011; David, 2011; Xue and Huang, 2023; David, 2007; Eckert et al., 2009; Straccia and Troncy, 2005; Euzenat et al., 2005).

As future work, we propose to enrich the alignment approach by adding another set of ontology elements, such as sub-classes and individuals. In addition, we will test other machine learning models and select the best performing model for the ontology alignment task.

## REFERENCES

- Bock, J., Dänschel, C., and Stumpp, M. (2011). Mappso and mapevo results for oaei 2011. *Ontology Matching*, 179(10.5555):2887541–2887559.
- Bulygin, L. (2018). Combining lexical and semantic similarity measures with machine learning approach for ontology and schema matching problem. In *Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2018)*, pages 245–249.
- Bulygin, L. and Stupnikov, S. A. (2019). Applying of machine learning techniques to combine string-based, language-based and structure-based similarity measures for ontology matching. In *DAMDID/RCDL*, pages 129–147.
- David, J. (2007). Association rule ontology matching approach. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(2):27–49.
- David, J. (2011). Aroma results for oaei 2011. *Ontology Matching*, 122.
- Eckert, K., Meilicke, C., and Stuckenschmidt, H. (2009). Improving ontology matching using meta-level learning. In *The Semantic Web: Research and Applications: 6th European Semantic Web Conference, ESWC 2009 Heraklion, Crete, Greece, May 31–June 4, 2009 Proceedings 6*, pages 158–172. Springer.
- Euzenat, J., Guégan, P., and Valtchev, P. (2005). Ola in the oaei 2005 alignment contest. In *Proc. K-Cap 2005 workshop on Integrating ontology*, pages 97–102. No commercial editor.
- Euzenat, J., Shvaiko, P., et al. (2007). *Ontology matching*, volume 18. Springer.
- Gruber, T. R. and Olsen, G. R. (1994). An ontology for engineering mathematics. In *Principles of Knowledge Representation and Reasoning*, pages 258–269. Elsevier.
- Huber, J., Szttyler, T., Noessner, J., and Meilicke, C. (2011). Codi: Combinatorial optimization for data integration—results for oaei 2011. *Ontology Matching*, 134.
- Ondřej Zamazal, Jana Vataščinová, L. Z. Initiative d'évaluation de l'alignement des ontologies - campagne oaei-2021.
- Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350.
- Shvaiko, P. and Euzenat, J. (2005). A survey of schema-based matching approaches. In *Journal on data semantics IV*, pages 146–171. Springer.
- Straccia, U. and Troncy, R. (2005). omap: Combining classifiers for aligning automatically owl ontologies. In *International Conference on Web Information Systems Engineering*, pages 133–147. Springer.
- Xue, X. and Huang, Q. (2023). Generative adversarial learning for optimizing ontology alignment. *Expert Systems*, 40(4):e12936.