Machine Learning for Ontology Alignment

Faten Abbassi[1][a], Yousra Bendaly Hlaoui[1][b] and Faouzi Ben Charrada[2][c]

[1] *LIPSIC Laboratory, University of Tunis El Manar Tunis, Tunisia*

[2]

faten.abbassi@fst.utm.tn, yousra.hlaoui@fst.utm.tn, faouzi.bencharrada@fst.utm.tn

Keywords: Machine Learning, Normalization Techniques, Reference Ontologies, Conference Track, Benchmark Track.

Abstract: This article proposes an ontology alignment approach that combines supervised machine learning models and schema-matching techniques. Our approach analyzes reference ontologies and their alignments provided by *OAEI* to extract ontological data matrices and confidence vectors. In addition, these ontological data matrices are normalized using normalization techniques to obtain a coherent format for enhancing the accuracy of the alignments. From the normalized data, syntactic and external similarity matrices are generated via individual matchers before being concatenated to build a final similarity matrix representing the correspondences between two ontologies. This matrix and the confidence vector are then used by six machine learning models, such as *Logistic Regression*, *Random Forest Classifier*, *Neural Network*, *Linear SVC*, *K-Neighbors Classifier* and *Gradient Boosting Classifier*, to identify ontological similarities. To evaluate the performances of our approach, we have compared our results with our previous results (Abbassi and Hlaoui, 2024a). The experiments are performed over the reference ontologies of the *benchmark* and *conference* tracks based on their reference alignments provided by *OAEI*.

1 INTRODUCTION

The heterogeneity of information representations in computer science results from the use of different vocabularies, concepts and structures, which makes interoperability between systems complex, particularly on the Internet. To overcome these challenges, ontologies have been proposed as a solution. An ontology is a formal and explicit specification of a shared conceptualization (Gruber, 1993; Gruber and Olsen, 1994). They unify different points of view by reducing or eliminating conceptual and terminological confusion (Uschold and Gruninger, 1996). Indeed, the same ontology conceptualizes the same knowledge that could be specified by different ontologies. The management of ontological diversity relies on ontology alignment, which aims to unify heterogeneous entities while conserving the coherence of information. An ontology alignment process consists of computing similarity measures between the different classes and links of a pair of ontologies (Euzenat et al., 2007). This process is performed by schema matching techniques, which calculate similarities at different levels of ontology granularity (Rahm and Bernstein, 2001; Euzenat et al., 2007; Shvaiko and Euzenat, 2005): element level (classes and individuals), internal structure level (data properties) and external structure level(subclasses, disjoint classes and relationships). Each of these techniques is implemented by individual matchers that compute similarity measures at different levels of ontology granularity (Rahm and Bernstein, 2001; Euzenat et al., 2007; Shvaiko and Euzenat, 2005) . In addition, the different individual matchers depend on how they generally interpret the input information(Rahm and Bernstein, 2001; Euzenat et al., 2007; Shvaiko and Euzenat, 2005). Each of them calculates similarity measures according to syntactic interpretation criteria, where labels are treated as character strings, and/or external interpretation criteria, where labels are perceived as linguistic objects through external resources, such as a thesaurus. Schema-matching techniques are based not only on individual matching matchers but also on composite matchers. In fact, composite matchers combine ontology similarity measures obtained by different individual matchers to determine the final ontology alignment decision for a given pair of ontologies(Rahm and Bernstein, 2001; Euzenat et al., 2007; Shvaiko and Euzenat, 2005). We have opted

[a] https://orcid.org/0000-0001-7525-4505
[b] https://orcid.org/0000-0002-3476-0185
[c] https://orcid.org/0000-0001-5484-547X

Abbassi, F., Hlaoui, Y. B. and Ben Charrada, F.
Machine Learning for Ontology Alignment.
DOI: 10.5220/0013425700003928

In *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2025)*, pages 668-675
ISBN: 978-989-758-742-9; ISSN: 2184-4895

to use machine learning algorithms to automate composite matchers. This choice is justified by the efficiency and the potential of these algorithms, to enhance the accuracy of this task, presented in (Abbassi and Hlaoui, 2024a; Abbassi and Hlaoui, 2024b; Xue and Huang, 2023). Since we manipulate labeled data as well as continuous data representing the values of ontological similarity measures defined in the interval $[0,1]$, we have developed an approach based on supervised machine learning algorithms dedicated to classification.

Despite the large number of ontology alignment approaches based on supervised machine learning in the literature (Abbassi and Hlaoui, 2024a; Abbassi and Hlaoui, 2024b; Xue and Huang, 2023), these approaches have several limits, including:

- The use of a limited number of similarity measure techniques and machine learning models.
- Partial exploitation of the ontology structure in the alignment process reduces the accuracy of obtained results.
- Use of incorrect values for similarity measures, leading to erroneous results.
- Aligning classes separately from their subclasses, disjoint classes, data properties, object properties, individuals and comments, which has a negative impact on the accuracy of the alignment process.

To overcome these limits, we propose a new ontology alignment approach that extends our previous work (Abbassi and Hlaoui, 2024a) by integrating other ontological information, exploring new supervised machine learning models and comparing our results with our previous approach.

Given the current state of the literature, the contribution of the present work consists of :

- Testing a large number of supervised machine learning techniques to ensure better ontology alignment.
- Enrichment of the alignment approach by adding new ontological information.
- Alignment of entities according to ontology structure.

The remainder of this paper is structured as follows. Section **2** provides a review of related works. Section **3** presents our alignment approach in detail. Section **4** presents the experimentation and quality evaluation of our alignment approach. Section **5** presents a discussion of this paper. Finally, Section **6** concludes this paper and proposes future perspectives.

2 RELATED WORKS

In this study, we are particularly interested in approaches based on machine learning and schema-matching techniques (Abbassi and Hlaoui, 2024a; Abbassi and Hlaoui, 2024b; Xue and Huang, 2023). The approach proposed by (Abbassi and Hlaoui, 2024b) uses several machine learning models such as *LogisticRegression*, *GradientBoostingClassifier*, *GaussianNB*, and *KNeighborsClassifier* to perform ontology alignment. It uses 21 string-based similarity measures and three language-based similarity measures. These techniques are applied to class labels, data property labels and labels of relationships between classes. However, this approach has certain limits, notably (i) limited exploitation of the global ontological structure and (ii) lack of precision in some cases. The authors in (Abbassi and Hlaoui, 2024a) have developed an approach using 30 similarity measure techniques based on string and language aspects. These techniques are applied to class labels, sub-classes, data properties, relationships between classes and individual labels. It uses five machine learning models: *LogisticRegression*, *RandomForestClassifier*, *Neural Network*, *LinearSVC* and *GradientBoostingClassifier*. However, this approach has certain limits. It lacks precision for some ontology pairs and does not exploit the integral structure of ontologies. The approach described in (Xue and Huang, 2023) is based on syntactic similarity measure techniques such as the Levenshtein distance, the Jaro distance, the Dice coefficient, the N-gram and the WordNet language technique. It combines an unsupervised machine learning model, the Generative Adversarial Network, with the Simulated Annealing Algorithm (SA-GAN). However, this approach has an important limit: a lack of accuracy, mainly due to the non-respect of ontology structure. Specifically, the alignment process treats entities independently of their properties and subclasses, which reduces alignment accuracy.

3 PROPOSED APPROACH

According to figure1, this approach is mainly composed of five steps, namely *Ontology and Reference Alignment Parsing Step*, *Ontology Normalization Step*, *Ontology Similarity Value Computing Step*, *Final Similarity Matrix Construction Step*, *Ontology Training and Testing Step* and *Quality Evaluation of Ontology Alignment Step*.

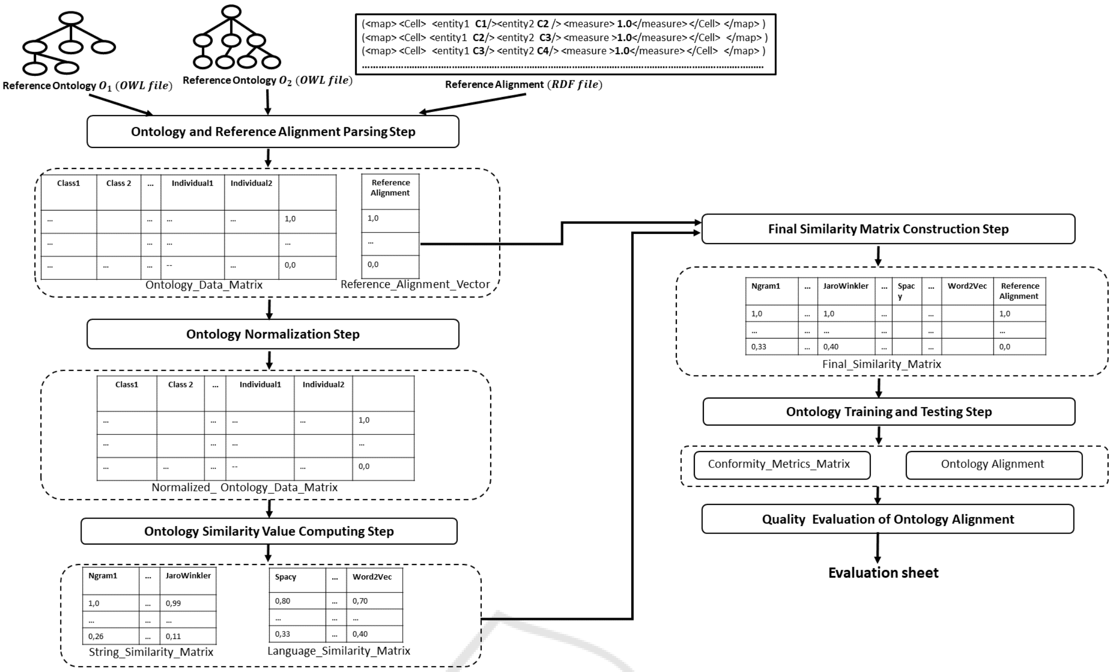
Reference Ontology O_1 ($OWL\ file$)
Reference Ontology O_2 ($OWL\ file$)
(<map> <Cell> <entity1 C1/><entity2 C2 /> <measure> 1.0</measure> </Cell> </map>)
(<map> <Cell> <entity1 C2/> <entity2 C3/> <measure >1.0</measure> </Cell> </map>)
(<map> <Cell> <entity1 C3/> <entity2 C4/> <measure >1.0</measure> </Cell> </map>)
Reference Alignment ($RDF\ file$)
Ontology and Reference Alignment Parsing Step
Class1
Class 2
Individual1
Individual2
Reference Alignment
Ontology_Data_Matrix
Reference_Alignment_Vector
Ontology Normalization Step
Normalized_ Ontology_Data_Matrix
Ontology Similarity Value Computing Step
Ngram1
JaroWinkler
String_Similarity_Matrix
Spacy
Word2Vec
Language_Similarity_Matrix
Final Similarity Matrix Construction Step
Final_Similarity_Matrix
Ontology Training and Testing Step
Conformity_Metrics_Matrix
Ontology Alignment
Quality Evaluation of Ontology Alignment
Evaluation sheet

Figure 1: Proposed Approach.

3.1 Step 1: Ontology and Reference Alignment Parsing

As Figure1 shows, this phase takes as input a pair of reference ontologies in the form of two OWL [1] files and their corresponding reference alignment in the form of an RDF [2] files, which are provided by *OAEI*. As output, it produces a matrix and a vector, which are essential for the next step in our alignment approach. The construction of the matrix consists of the extraction of ontological data from the input ontologies, while the construction of the vector is based on the extraction of the confidence values associated with these ontological data, calculated by the *OAEI* from the corresponding reference alignment file. The extracted ontological data includes *class labels*, *sub-class labels*, *disjoint class labels*, *data property labels (dataProperty)*, *relationship labels between classes (objectProperty)*, *individual labels* and *class comments*. This phase is implemented by ***Data_Construction*** algorithm 1, which generates the following results:

- The ***Ontology_Data_Matrix***, where each row represents an instance of a *VData_Ontology* vector. Each *VData_Ontology* vector contains the ontological data extracted from the input ontology pair.
- The ***Reference_Alignment_Vector***, which contains the confidence values (calculated by *OAEI*) associated with the data in the ontology data matrix from the reference alignment file.

Thus, each of the product matrix and the vector has a number of rows equal to $|ClassesO1 \times ClassesO2|$. Where $|ClassesO1 \times ClassesO2|$ represents the cardinality of the Cartesian product of ontology classes $O1$ and ontology classes $O2$.

3.2 Step 2: Ontology Normalization

The objective of this step is to clear and transform the ontological data extracted in the first phase into a common format, in order to enhance the alignment results. This step takes as input the ontological data matrix ($Ontology\_Data\_Matrix$) produced in the previous step and generates, as output, a normalized data matrix ($Normalized\_Ontology\_Data\_Matrix$) required for the next step. In fact, we have applied diverse normalization techniques to class labels, sub-class labels, disjoint class labels, data property labels (dataProperty), relationship labels between classes (objectProperty), individual labels and comments associated to classes. The used normalization techniques include: *case normalization* technique, *blank normalization* technique, *link stripping* technique, *punctuation elimination*, *diacritics suppression* technique and *digit suppression* technique.

[1] https://www.w3.org/OWL/

[2] https://www.w3.org/RDF/

Algorithm 1: Data_Construction.

```
Data: O1.owl, O2.owl, Reference_Alignment.rdf;
Result: Ontology_Data_Matrix: array of VData_Ontology
        vector (CSV file);
Reference_Alignment_Vector: Vector of reference alignment;
i ← 0;
for class, ClassO1, of O1.OWL do
    for class, ClassO2, of O2.OWL do
        Sub-Class1←Sub-classes Of ClassO1;
        Sub-Class2←Sub-classes Of ClassO2;
        Disjoint-Class1←Disjoint-classes Of ClassO1;
        Disjoint-Class2←Disjoint-classes Of ClassO2;
        Data-Propclass1 ← Data Properties Of ClassO1;
        Data-Propclass2 ← Data Properties Of ClassO2;
        Object-Propclass1 ← Object Properties Of ClassO1;
        Object-Propclass2 ← Object Properties Of ClassO2;
        Individuals-class1 ← individuals Of ClassO1;
        Individuals-class2 ← individuals Of ClassO2;
        Comments-class1 ← Comments Of ClassO1;
        Comments-class2 ← Comments Of ClassO2;
        VData_Ontology←(ClassO1, ClassO1, Sub − Class1,
          Sub − Class2, Disjoint − Class1, Disjoint − Class2,
          Data − Propclass1, Data − Propclass1,
          Object − Propclass1, Object − Propclass2,
          Individuals − class1, Individuals − class2,
          Comments − class1, Comments − class2);
        Ontology_Data_Matrix[i]← VData_Ontology;
        if (ClassO1, ClassO2 ) ∈ Reference_Alignment.rdf
          then
            Confident_alignment ← 1 ;
        else
            Confident_alignment ← 0 ;
        end
        Reference_Alignment_Vector[i]←
          Confident_alignment;
        i ← i+1;
    end
end
```

3.3 Step 3: Ontology Similarity Value Computing

This step takes as input the normalized ontology data matrix produced at the end of the *Ontology Normalization* step. It aims to calculate the syntactic and external similarity measures for the different entity pairs stored in the input normalized ontology data matrix. As output, the current step generates two matrices: a *String_Similarity_Matrix* and a *Language_Similarity_Matrix*. These matrices contain the syntactic and external similarity values calculated for the ontological data stored in the input matrix.

To build the syntactic similarity matrix, we used 26 individual matchers implementing 26 string-based techniques (Abbassi and Hlaoui, 2024a)(see Table 1). For the construction of the external similarity matrix, we have used four individual matchers implementing four language-based techniques(see Table 1).

Algorithm 2: Calculation_Similarity_Values.

```
Data: Normalized_Ontology_Data_Matrix: matrix of
      Normalized VData_Ontology vectors;
Result: String_Similarity_Matrix, Language_similarity_Matrix :
        matrices of real values (CSV file);
i ← 0;
for VDATA of Normalised_Ontology_Data_Matrix do
    for each information of VDATA do
        SVsim_Class←String_Sim_Class
          (NClassO1,NClassO2);
        SVsim_SubClass←String_Sim_SubClass
          (NSubClassO1,NSubClassO2);
        SVsim_DisjointClass←String_Sim_DisjointClass
          (NDisjointClassO1, NDisjointClassO2);
        SVsim_DataProperties←String_Sim_DataProperties
          (NDataPropertiesO1,NDataPropertiesO2);
        SVsim_ObjectProperties←String_Sim_ObjectProperties
          (NObjectPropertiesO1,NObjectPropertiesO2);
        SVsim_Individuals←String_Sim_Individuals
          (NIndividualsO1,NIndividualsO2);
        SVsim_ClassComments←String_Sim_ClassComments
          (NClassCommentsO1,NClassCommentsO2);
        Syntactic_VSim←GlobalConstructor(SVsim_Class,
          SVsim_SubClass, SVsim_DisjointClass,
          SVsim_DataProperties, SVsim_ObjectProperties,
          SVsim_Individuals, SVsim_ClassComments);
        String_Similarity_Matrix [i]←Syntactic_VSim ;
        i ← i+1;
    end
end
j ← 0;
for VDATA of Normalised_Ontology_Data_Matrix do
    for each information of VDATA do
        LVsim_Class←Language_Sim_Class
          (NClassO1,NClassO2);
        LVsim_SubClass←Language_Sim_SubClass
          (NSubClassO1,NSubClassO2);
        LVsim_DisjointClass←Language_Sim_DisjointClass
          (NDisjointClassO1,NDisjointClassO2);
        LVsim_DataProperties←Language_Sim_DataProperties
          (NDataPropertiesO1,NDataPropertiesO2);
        LVsim_ObjectProperties←Language_Sim_ObjectProperties
          (NObjectPropertiesO1,NObjectPropertiesO2);
        LVsim_Individuals←Language_Sim_Individuals
          (NIndividualsO1,NIndividualsO2);
        LVsim_ClassComments←Language_Sim_ClassComments
          (NClassCommentsO1,NClassCommentsO2);
        Language_VSim←GlobalConstructor(LVsim_Class,
          LVsim_SubClass, LVsim_DisjointClass,
          LVsim_DataProperties, LVsim_ObjectProperties,
          LVsim_Individuals, LVsim_ClassComments);
        Language_Similarity_Matrix [j]←Language_VSim ;
        j ← j+1;
    end
end
```

These techniques are applied to each pair of elements stored in the input ontology data matrix. They include the pairs of classes, the pairs of subclasses, the pairs of disjoint classes, the pairs of data proper-

Table 1: Used String and language based techniques.

Technique class	Techniques
String-based techniques	N-gram 1, N-gram 2, N-gram 3, N-gram 4, Dice coefficient, Jaccard similarity, Jaro measure, Monge-Elkan, Smith-Waterman, Needleman-Wunsh, Affine gap, Bag distance, Cosine similarity, Partial Ratio, Soft TF-IDF, Editex, Generalized Jaccard, Jaro-Winkler, Levenshtein distance, Partial Token Sort Fuzzy Wuzzy Ratio, Soundex, TF-IDF, Token Sort, TverskyIndex, Overlap coefficient (Euzenat et al., 2007; Abbassi and Hlaoui, 2024a).
Language-based techniques	Wu and Palmer similarity,Word2vec, Sentence2vec similarity and Spacy (Euzenat et al., 2007; Abbassi and Hlaoui, 2024a).

ties (*dataProperty*), the pairs of relationships between classes (*objectProperty*), the pairs of individuals and the pairs of class comments. The current step is implemented by the ***Calculation_Similarity_Values*** algorithm(cf. Algorithm 2), which uses the following functions:

1. ***String_Sim_Entity*** to create and return syntactic similarity vectors for each pair of ontological entities stored in the ***Normalized_Ontology_Data_Matrix***.
2. ***Language_Sim_Entity*** to create and return external similarity vectors for each pair of entities in the ***Normalized_Ontology_Data_Matrix***.
3. ***GlobalConstructor*** to concatenate the syntactic and external similarity vectors and produce the Syntactic_VSim and Language_VSim vectors for each pair of classes of the pair of ontologies to be aligned. As a result, the Syntactic_VSim vector contains 182 similarity values (26 x 7), while the Language_VSim vector contains 28 (4 x 7) similarity values.

Indeed, each of generated matrices has a number of rows equal to the cardinality of the cartesian product of ontology $O1$ classes and ontology $O2$ classes, i.e. $|ClassesO1 \times ClassesO2|$.

3.4 Step 4: Final Similarity Matrix Construction

This step takes as input the similarity matrices generated by the previous step and the $Reference\_Alignment\_Vector$ produced by the ***Ontologies and Reference Alignment Parsing*** step. The objective of this step is to combine these matrices with the reference vector to obtain a final similarity matrix. Hence, the construction of this $Final\_Similarity\_Matrix$ proceeds as follows:

- **Step 1:** Each row of the syntactic similarity matrix, representing an instance of the $Syntactic\_VSim$ vector, each row of the external similarity matrix, representing an instance of the $Language\_VSim$ vector, and each row of the $Reference\_Alignment\_Vector$ are concatenated to form a final similarity vector, named $Final\_VSim$. This vector contains all the calculated similarity values and the reference alignment of each pair of elements concerned. It is defined as follows:

$$\texttt{Final\_VSim} = (\texttt{Syntactic\_VSim}, \texttt{Language\_VSim} \\ , \texttt{ConfidentAlignement})$$

This vector contains 210 similarity values (resulting from the concatenation of the of the $Syntactic\_VSim$ and the $Language\_VSim$ vectors) and integrates the reference alignment for the corresponding pair of classes.

- **Step 2:** The $Final\_Similarity\_Matrix$ is then constructed, where each row represents an instance of the calculated $Final\_VSim$. This matrix has the same number of rows as the similarity matrices created at the end of the previous step.

3.5 Step 5: Ontology Training and Testing

The role of this phase is to produce the final alignment result between a given pair of ontologies. It takes as input the $Final\_Similarity\_Matrix$ generated by the previous step and six machine learning models. The selected models are *Logistic Regression*, *Random Forest Classifier*, *Neural Network*, *Linear SVC*, *K-Neighbors Classifier* and *Gradient Boosting Classifier*, which are the most frequently used in the literature (Bento et al., 2020; Abbassi and Hlaoui, 2024a; Xue and Huang, 2023). As an output, this step provides the degree of similarity of the input ontology pair and generates a matrix of conformity metrics. This matrix includes the precision (P), recall (R) and f-measure , calculated by each used machine learning model (Euzenat et al., 2007; Abbassi and Hlaoui, 2024a). These measures are defined as follows:

$$P : \Lambda \times \Lambda \to [0..1] \quad R : \Lambda \times \Lambda \to [0..1]$$

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

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

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

The training and testing process for each used machine learning model is detailed as follows:

- **Step 1 Model Training:** In this first step, we build a training matrix consisting of 60% of the rows of the final similarity value matrix for a pair of ontologies to be aligned. Each machine learning model is then trained from this matrix, using the first 210 columns to build their necessary datasets for evaluating the degree of similarity between the ontologies. This creates trained models capable of capturing the relationships between ontological entities.
- **Step 2 Model Testing:**: This stage consists of preparing a test matrix containing the remaining 40% of the rows of $Final\_Similarity\_Matrix$ for the same pair of ontologies. Each trained model is tested based on the first 210 columns of this test matrix, to produce the final alignment for the current pair of ontologies. This test evaluates each model's capacity to generalize and predict alignments for new data.
- **Step 3 Model Evaluation:** Finally, each machine learning model is evaluated using conformity metrics, which measure the degree of correspondence between the degrees of similarity predicted by the models and the confidence alignment values provided in the $ReferenceAlignment$ column. This evaluation is used to determine the accuracy and efficiency of the models in establishing alignments between ontologies.

We have implemented the training and testing process for each pair of reference ontologies and for the six machine learning models that we have used. As a result, we have generated 29 matrices of conformity metrics.

3.6 Step 6: Quality Evaluation of Ontology Alignment

The main objective of this step is to validate the results obtained by the different machine learning models that we have used. This validation is based on the matrices of conformity metrics generated by the previous step, specifically focusing on the *f-measure* metric provided by each model for the set of alignment tests performed on the reference ontology pairs. This choice is justified because the *f-measure* calculates the harmonic mean of precision and recall, giving them equivalent importance (Euzenat et al., 2007; Abbassi and Hlaoui, 2024a). This validation consists of comparing the results provided by our approach with those of our previous approach (Abbassi and Hlaoui, 2024a).

4 EXPERIMENTATION AND QUALITY EVALUATION OF THE ALIGNMENT APPROACH

To implement the different steps of our approach, we have configured a work environment using *Anaconda 1.10.1* and the *Spyder* editor, specially designed for Python development. This configuration includes the installation and the use of tools such as the $py\_stringmatching$, the $beautifulsoup4$, the $Owlready2$, the $pandas$, the $fuzzycomp$, the $NGram$, the $WordNet$, the $nltk(NaturalLanguageToolkit)$, the $spacy$, the $en\_core\_web\_lg$, the $Gensim$, the $tqdm$, the $Keras$ and the $sklearn$ libraries, as well as the $GoogleNews - vectors - negative3 - 0011$ Dictionary. These tools are running on a laptop with Windows 10 Professional N 64-bit operating system, Intel Core i7-8550U processor (1.80 GHz - 1.99 GHz) and 8 GB RAM. To configure the hyper parameters of the different machine learning models that we have used, we have employed those described in Table 2. To experiment our approach, we have focused on the *benchmark* [3] and *conference* [4] tracks, which are frequently used in the literature. Each includes reference ontologies in *OWL* file format and their reference alignments in *RDF* file format. The reference track comprises various ontologies modified according to three test families: the 1xx family, the 2xx family and the 3xx family. The *conference* track presents the highest degree of heterogeneity compared with the other tracks. It includes seven reference ontologies, generating 21 pairs of ontologies with their reference alignments. We have used eight cases of alignment of the *benchmark* track and 21 cases of alignment of the *conférence* track.

4.1 Quality Evaluation of the Current Ontology Alignment Approach

The validation of the results provided by our alignment approach applied to the different tested machine

[3] https://oaei.ontologymatching.org/2016/benchmarks/

[4] http://oaei.ontologymatching.org/2024/conference/

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

Machine Learning Model	**Hyper parameters**
Random Forest Classifier	n_estimators=100, max_features=None, max_depth=2
Neural Network	Dense layer = 211 neurons with Activation function = relu. Dense output layer 1 neuron with activation function sigmoid. Compiler of the algorithm with an adam optimizer. The Metric accuracy. The binary crossentropy loss function.
Linear SVC	C=0.5, penalty="l2", dual=False
K-Neighbors Classifier	n_neighbors=1
Gradient Boosting Classifier	learning rate = 1, n_estimators= 100
Logistic Regression	max iter= 1000, solver='lbfgs'

Table 3: Values of f-measure of our approach compared to our previous approach (Abbassi and Hlaoui, 2024a) for each pair of ontology tests in the 2024 OAEI conference track.

Pair of Reference Ontologies	**Our Approach**						**Our previous Approach (Abbassi and Hlaoui, 2024a)**				
	LR	**RFC**	**NN**	**LSVC**	**KNC**	**GBC**	**XGB**	**NN**	**LSVC**	**LR**	**RFC**
cmt-conference	0.66	0.40	0.88	0.40	0.52	0.57	0.44	0.80	0.44	0.44	0.40
cmt-confOf	0.60	0.50	0.64	0.62	0.50	0.66	0.53	0.53	0.53	0.53	0.57
cmt-edas	0.90	0.97	1.00	1.00	0.97	1.00	1.00	1.00	1.00	1.00	0.85
cmt-ekaw	0.74	0.82	0.90	0.88	0.86	0.85	0.76	0.76	0.76	0.76	0.76
cmt-iasted	0.64	0.60	1.00	0.97	0.90	0.80	0.88	1.00	0.88	0.88	0.57
cmt-sigkdd	0.88	0.77	0.97	0.86	0.81	0.96	0.88	0.88	0.88	0.82	0.62
conference-confOf	0.88	0.85	0.90	0.88	0.86	0.88	0.73	0.80	0.73	0.73	0.73
conference-edas	0.53	0.51	0.77	0.77	0.74	0.61	0.63	0.66	0.63	0.63	0.47
conference-ekaw	0.58	0.51	0.40	0.53	0.50	0.60	0.50	0.47	0.45	0.45	0.50
conference-iasted	0.50	0.44	0.51	0.40	0.44	0.53	0.44	0.55	0.44	0.52	0.44
conference-sigkdd	0.85	0.88	0.85	0.85	0.81	0.85	0.76	0.76	0.76	0.76	0.76
confOf-edas	0.66	0.66	0.77	0.71	0.76	0.77	0.69	0.69	0.69	0.63	0.57
confOf-ekaw	0.63	0.77	0.74	0.62	0.60	0.57	0.60	0.60	0.55	0.55	0.64
confOf-iasted	0.66	0.74	0.77	0.73	0.70	0.76	0.61	0.61	0.61	0.61	0.61
confOf-sigkdd	0.80	0.90	0.96	0.88	0.81	0.88	0.80	0.80	0.80	0.80	0.80
edas-ekaw	0.76	0.77	0.76	0.70	0.69	0.76	0.62	0.64	0.64	0.68	0.68
edas-iasted	0.53	0.63	0.66	0.60	0.54	0.44	0.51	0.51	0.51	0.46	0.51
edas-sigkdd	0.88	0.85	0.86	0.81	0.86	0.88	0.77	0.77	0.77	0.77	0.70
ekaw-iasted	0.74	0.88	0.90	0.88	0.80	0.90	0.74	0.75	0.74	0.66	0.74
ekaw-sigkdd	0.81	0.90	0.96	0.88	0.85	0.81	0.77	0.77	0.77	0.77	0.77
iasted-sigkdd	1.00	0.90	1.00	0.97	0.96	0.96	0.85	0.81	0.81	1.00	0.85

LR: ***LogisticRegression***, RFC: ***RandomForest***, GNB: ***GaussianNB***, NN: ***Neural Network***, LSVC: ***Linear SVC***, KNC: ***KNeighborsClassifier***, GBC: ***GradientBoostingClassifier***, XGB: ***XGBoost***.

Table 4: Values of f-measure of our approach compared to our previous approach (Abbassi and Hlaoui, 2024a) for eight pairs of ontology tests in the benchmark track.

Pair of Reference Ontologies	**Our previous approach (Abbassi and Hlaoui, 2024a)**					**Our Approach**					
	RFC	**NN**	**LSVC**	**LR**	**GBC**	**LR**	**RFC**	**NN**	**LSVC**	**KNC**	**GBC**
201-208	0.79	0.81	0.85	0.83	0.75	0.80	0.88	0.92	0.96	0.74	0.86
221-247	0.89	0.95	0.96	0.93	0.92	0.97	0.90	1.00	1.00	0.96	0.97
301-304	0.80	0.88	0.95	0.80	0.82	0.88	0.85	0.85	0.96	0.77	0.90
248-266	0.50	0.55	0.59	0.52	0.52	0.66	0.66	0.77	0.74	0.69	0.74
101-104	0.96	1.00	1.00	1.00	0.98	1.00	0.96	1.00	1.00	0.88	1.00
101-302	0.97	0.90	1.00	0.92	0.92	0.96	0.95	0.95	1.00	0.74	0.97
101-303	0.97	0.93	1.00	0.92	0.88	1.00	1.00	1.00	1.00	0.90	0.90
101-304	0.98	1.00	0.97	0.99	1.00	1.00	1.00	1.00	1.00	0.97	1.00

LR: ***LogisticRegression***, GBC: ***GradientBoostingClassifier***, KNC: ***KNeighborsClassifier***, NN: ***Neural Network***, RFC: ***RandomForest***, LSVC: ***Linear SVC***.

learning models is based on comparing the results obtained and those of our previous approach (Abbassi and Hlaoui, 2024a). This comparison is based on the *f-measure* metric (see section 3.5), applied to all used models (see Tables 3 and 4). The choice of the f-measure is justified by its capacity to calculate the harmonic mean of precision and recall according to them equal importance (Euzenat et al., 2007; Abbassi and Hlaoui, 2024a)(see section 3.5). Then, these results show a percentage higher than **70%** up to **90%** for the *conference* track and a percentage higher than **60%** up to **100%** for the *benchmark* track. These percentages represent the better performance achieved by our approach compared to the previous approach. This confirms the enhanced quality and performance of our results compared to those obtained by our previous results (Abbassi and Hlaoui, 2024a). This performance is due to the construction of ontological data matrices, respecting the structure of ontologies, which was crucial in improving alignment results. Although the same machine learning models are used, with the addition of the *Neighbors Classifier* model, the accuracy of the results is improved through richer data matrices. However, some special cases, such as the ontology pairs *101-104*, *221-247*, *cmt-conference*, *conference-ekaw*, *conference-iasted* and *edas-iasted*, show lower accuracy. This is often due to the structure and absence of certain ontological elements, such as disjoint classes or comments.

5 CONCLUSION

This paper presents an ontology alignment method based on supervised machine learning and automatic schema-matching. Our approach follows a series of successive steps: ontology and reference alignment parsing step, ontology normalization step, ontology similarity value computing step, final similarity matrix construction step, ontology training and testing step, and quality evaluation of ontology alignment step. The first step consists to analyze the reference ontologies and their alignments provided by the *OAEI* to extract ontological data matrices and confidence vectors. This forms a basis for further processing. Then, these data are normalized into a coherent format, which enhances the accuracy of the alignments. Syntactic and external similarity matrices are subsequently generated by individual matches applied to the normalized data and then merged to create a final similarity matrix representing the correspondence between a pair of ontologies. This matrix is exploited by machine learning models to identify ontological similarities, and the quality of our alignment is evaluated by comparing our results with those of our previous results (Abbassi and Hlaoui, 2024a). Our experiments indicate that our approach improves accuracy over previously published methods.

As future work, we propose enriching the alignment approach by adding other ontological entities, such as ontology comments, equivalent classes, super-classes and equivalent individuals of a given class.

REFERENCES

Abbassi, F. and Hlaoui, Y. B. (2024a). An ontology alignment validation approach based on supervised machine learning algorithms and automatic schema matching approach. In *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 332–341. IEEE.

Abbassi, F. and Hlaoui, Y. B. (2024b). Supervised machine learning models and schema matching techniques for ontology alignment.

Bento, A., Zouaq, A., and Gagnon, M. (2020). Ontology matching using convolutional neural networks. In *Proceedings of the 12th language resources and evaluation conference*, pages 5648–5653.

Euzenat, J., Shvaiko, P., et al. (2007). *Ontology matching*, volume 18. Springer.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.

Gruber, T. R. and Olsen, G. R. (1994). An ontology for engineering mathematics. In *Principles of Knowledge Representation and Reasoning*, pages 258–269. Elsevier.

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.

Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(2):93–136.

Xue, X. and Huang, Q. (2023). Generative adversarial learning for optimizing ontology alignment. *Expert Systems*, 40(4):e12936.