

# Pushing the Boundaries: Classification of Entity Alignment from RDF Embeddings

Bill Gates Happi Happi <sup>a,b,\*</sup>, Géraud Fokou Pelap <sup>c</sup>, Danai Symeonidou <sup>d</sup> and Pierre Larmande <sup>a,b</sup>

<sup>a</sup> *LIRMM, Department of Computer Science, University of Montpellier, Montpellier, 34095, France*

*E-mail: pierre.larmande@ird.fr*

<sup>b</sup> *DIADE, IRD, CIRAD, University of Montpellier, Montpellier, 34394, France*

*E-mail: bill.happi@ird.fr*

<sup>c</sup> *URIFIA, Department of Computer Science, University of Dschang, Dschang, Cameroon*

*E-mail: geraud.fokou@univ-dschang.org*

<sup>d</sup> *Mistea, INRAE, 34060, Montpellier, France*

*E-mail: danai.symeonidou@inrae.fr*

**Abstract.** Entity Alignment (EA) involves identifying entities across two knowledge bases that represent the same real-world entity. This task is crucial for the automated integration of multiple Knowledge Graphs (KG) thus enriching the knowledge. Recently, embedding methods based on KG have become predominant in EA techniques. These methods project entities into a lower-dimensional space and align them by evaluating their similarities. However, the classification and alignment of entities between two KG remain complex. This article evaluates the performance of various classifiers across multiple aspects of entity embedding features, applicable to both source and target data in binary classification processes for EA. Our experiments indicate a consistent range in the F1-score and accuracy, particularly when dealing with imbalanced data and changes in the dimensions of embeddings. This observation suggests that future research may need to focus on developing more robust classification algorithms.

**Keywords:** Knowledge graph, Alignment, Embeddings, Classification, RDF, Similarities

## 1. Introduction

The foundational principle revolves around knowledge, which asserts that for a program to succeed at performing intricate tasks, it must possess an extensive understanding of the environment in which it operates [16]. Algorithms need typed data to work. Regarding Knowledge Graphs (KG), RDF (Resource Description Framework) graphs represent and structure complex knowledge through subject-predicate-object triplets. They are widely used for semantic data management in various domains. However, classifying Entity Alignments (EA) as referring to the same reality or not between two distinct RDF graphs remain a complex challenge. Approaches for calculating the numerical vectors associated with the entities in these KGs by aggregating their meaning in mutual relationships (symmetric, anti-symmetric, transitive), known as embedding, have made it easier to process these KGs using other numerical processing approaches. Some EA approaches still rely on manual calculations of fixed-size features combined with classification methods to carry out the process successfully [17]. Although embedding methods have shown notable effectiveness in generating features [11], it remains uncertain whether classification approaches can sustain their

---

\*Corresponding author. E-mail: bill.happi@ird.fr.

performance levels when applied to such embeddings. This study explores this potential limitation by assessing a variety of classification methods, including Logistic Regression [1], Support Vector Machine (SVM) [2], Decision Trees [3], Random Forest [4], Gradient Boosting [5], AdaBoost [6], Naive Bayes [7], K-Nearest Neighbors (KNN) [8], Artificial Neural Networks (ANN) [9], and Convolutional Neural Networks (CNN) [10]. These methods are evaluated based on concatenated entity embeddings derived from two distinct RDF graphs. More often, the embeddings used namely RDF2Vec [11], ComplEx-KG [12], RotatE [13], TransD [14], and others like them are known at capturing the features of entities within RDF graphs, regardless of differences in their structures and schemas. Especially in this study, we used RDF2Vec [11] because its logic is based on the word2vec[40] algorithm, which has so far contributed to the ability of machines to model the written language used by human beings to communicate in real life. Our study can be applied in data integration, document modeling, information retrieval, and knowledge discovery[15] to confirm the need to continue research into new classification approaches that are much more robust than existing ones. Our study will encompass the following key points :

- Embedding datasets whose entities need alignment;
- Preparing training and test sets by concatenating vectors derived from embedding dataset entities;
- Conducting a binary evaluation of these features using ten distinct classification approaches.

In the following sections, we will detail the related work and our methodology, present the results of our comparative analysis of classification methods, and discuss their implications for more effective use of semantic data and knowledge analysis.

## 2. Related work

In the Introduction (section 1), we present embedding approaches that aim to build numerical vectors for each entity from a KG. Numerous techniques lead to accomplishing this result as described in the review of Nezhadi et al [17]. These approaches show limits that could impact the following process [11]. Classifiers were made to help humans to solve decision problems in terms of classifications. Justo et al [42] recapitulated some algorithm families among which we have namely information-based, similarity-based, probability-based, and error-based learning. The tables 1 and 2 summarize the top 10 current classification approaches, highlighting their strengths, weaknesses, and associated entity alignment approaches. AdaBoost [18] and ExtraTrees [22] share a common feature, which is resistance to overfitting. K-Nearest Neighbors [25] and Gaussian Naive Bayes [27] are simple and intuitive in terms of strengths. K-Nearest Neighbors [25] does not necessarily require training data, while Gaussian Naive Bayes [27] often produces effective results on small datasets, unlike ExtraTrees [22] on large datasets. Random Forest [19], XGBoost [20, 21], and Gradient Boosting [28] highlight a strong ability to provide a classification model that takes into account outliers.

On the other hand, these approaches could have weaknesses that would impact their ability to ensure accurate predictions in real-life situations. Sensitivity to noisy data, which makes AdaBoost [18] and Gradient Boosting [28] vulnerable, does not guarantee high reliability in real-life conditions. The inability of some classifiers (Random Forest [19], XGBoost [20, 21], ExtraTrees [22], Logistic Regression [23], Decision Tree) to handle outliers can, in one way or another, lead to overfitting and may underperform on datasets. The daily observed data volume can also influence the processing time of certain classifiers (K-Nearest Neighbors [25], Gradient Boosting [28], ExtraTrees [22]), although they perform well in the absence of noise, outliers, or irrelevant features.

Entity alignment approaches exploited these classifiers, and their performance has been more or less satisfactory due to the dispersion of features generated by data vectorization approaches intended for classification algorithms. Sometimes, entity alignment approaches (columns “evaluate” in Tables 1 and 2) use mechanisms to help classifiers produce a consistent and reliable model for the classification task. Although aware of the added computation time by these approaches, it is clear that doubt could arise in situations of imminent decision-making and pose significant risks for a wrong choice, even though they often achieve an F1-score of up to 0.99. The optimal choice depends on the specific nature of the problem and data characteristics, underscoring the importance of a thorough understanding for informed decisions in real-world scenarios.

Bujang, S. et al [29] reviews methods for predicting student success in higher education, emphasizing imbalanced classification issues and recommending increased application of hybrid methods for improved predictive model generalization in student grade prediction. The diverse and varied nature of data should encourage researchers to stay motivated in developing new classification approaches.

### 3. Methodology

These methodological steps allowed us to conduct an in-depth analysis of entity alignment classification based on embeddings in the context of RDF graphs. In this study, we followed several crucial steps for the classification of entity alignments from source and target RDF data :

#### 3.1. Preliminary :

Consider an RDF data source (a graph) as a set of triplets defined by:  $G = \{(s, p, o) \in (R \cup B) \times (R) \times (L)\}$ , where  $R$  is the set of all resources in the form of IRIs (Internationalized Resource Identifier),  $B$  is the set of blank nodes, and  $L$  is the set of literals. We denote the set of RDF data sources as  $D$ . Let  $S(G)$ ,  $P(G)$ , and  $O(G)$  be the respective sets of subjects, predicates, and objects of a given data set  $G$ . Let  $\mathbb{N}$  be the set of natural numbers, including 0. Consider also  $i, j, a, b, k, l, m, n, v_{max}, u_{max} \in \mathbb{N}$ , which may or may not depend on the expression of the function they define as well as the application context. Let  $G_1$  be the set of source data containing  $m$  triplets, and  $G_2$  be the set of target data containing  $n$  triplets. We also consider the set  $V_{data} = \{(s_a, owl:sameAs, t_b); s_a \in S(G_1), t_b \in S(G_2) \text{ with } G_1 \text{ and } G_2 \in D\}$ , where  $s_a$  and  $t_b$  are any alignable entities from their respective sets.  $V_{data}$  is the set of valid data containing fundamental truths ( $V_{data} \subset D$  and  $owl:sameAs$  is the entity linking predicate). Subsequently, the notation  $(s_a, t_b, 1)$  is used interchangeably with  $(s_a, owl:sameAs, t_b)$ . The embedding function  $E : S(G) \rightarrow \mathbb{R}^d$  takes an element  $s \in S(G)$  as input and maps it to a  $d$ -dimensional vector in the real space  $\mathbb{R}^d$ .  $E(s) = \mathbf{v} (\mathbf{v} \in \mathbb{R}^d)$ .

Classification Approach	Strengths	Weaknesses	Evaluate
AdaBoost [18]	<ul style="list-style-type: none"> <li>– Resistant to overfitting.</li> <li>– Can be used with various types of weak classifiers.</li> </ul>	<ul style="list-style-type: none"> <li>– Sensitive to noisy data.</li> <li>– Less effective on data with a lot of noise.</li> </ul>	[30]
Random Forest [19]	<ul style="list-style-type: none"> <li>– Excellent performance on large datasets.</li> <li>– Robust to noisy data and outliers.</li> </ul>	<ul style="list-style-type: none"> <li>– Less interpretable than individual models.</li> <li>– Can overfit on complex datasets.</li> </ul>	[31]
XGBoost [20, 21]	<ul style="list-style-type: none"> <li>– High performance and accuracy.</li> <li>– Efficient handling of missing data and outliers.</li> </ul>	<ul style="list-style-type: none"> <li>– May require fine-tuning of hyperparameters.</li> <li>– Requires careful attention to avoid overfitting.</li> </ul>	[32]
ExtraTrees [22]	<ul style="list-style-type: none"> <li>– Robust to overfitting.</li> <li>– Can handle large amounts of data.</li> </ul>	<ul style="list-style-type: none"> <li>– Computationally expensive.</li> <li>– May not perform well on small datasets.</li> </ul>	[33]
Logistic Regression [23]	<ul style="list-style-type: none"> <li>– Simple and interpretable.</li> <li>– Efficient for linearly separable data.</li> </ul>	<ul style="list-style-type: none"> <li>– May underperform on complex, nonlinear data.</li> <li>– Sensitive to outliers.</li> </ul>	[34]

Table 1

Strengths and weaknesses of classification approaches on imbalanced data.

Classification Approach	Strengths	Weaknesses	Evaluate
Support Vector (SVC) [24]	<ul style="list-style-type: none"> <li>– Effective in high-dimensional spaces.</li> <li>– Versatile due to different kernel functions.</li> </ul>	<ul style="list-style-type: none"> <li>– Memory-intensive for large datasets.</li> <li>– Sensitive to choice of kernel and parameters.</li> </ul>	[35]
K-Nearest Neighbors (KNN) [25]	<ul style="list-style-type: none"> <li>– Simple and intuitive.</li> <li>– No training phase.</li> </ul>	<ul style="list-style-type: none"> <li>– Computationally expensive on large datasets.</li> <li>– Sensitive to irrelevant features.</li> </ul>	[36]
Decision Tree [26]	<ul style="list-style-type: none"> <li>– Easy to understand and interpret.</li> <li>– Can handle both numerical and categorical data.</li> </ul>	<ul style="list-style-type: none"> <li>– Prone to overfitting.</li> <li>– Can be sensitive to small variations in the data.</li> </ul>	[37]
Gaussian Naive Bayes [27]	<ul style="list-style-type: none"> <li>– Simple and computationally efficient.</li> <li>– Can perform well on small datasets.</li> </ul>	<ul style="list-style-type: none"> <li>– Assumes independence of features.</li> <li>– May not capture complex relationships in the data.</li> </ul>	[38]
Gradient Boosting [28]	<ul style="list-style-type: none"> <li>– High predictive accuracy.</li> <li>– Robust to outliers.</li> </ul>	<ul style="list-style-type: none"> <li>– Prone to overfitting, especially on noisy data.</li> <li>– Computationally expensive.</li> </ul>	[39]

Table 2

Strengths and weaknesses of classification approaches on imbalanced data.

### 3.2. Transformation into Embeddings :

We converted the entities of source and target RDF data into embeddings, and vectorial representations, using the RDF2Vec embedding method (see P2, Figure 1). RDF2Vec is a machine-learning method that generates vector representations for entities and relations in RDF graphs. Its advantages lie in its ability to capture the semantics of knowledge contained in these graphs, facilitating the search for similar entities and the analysis of complex relations. It is flexible and can be applied to diverse RDF graphs while remaining scalable to handle large datasets. In mathematical notation, we have:

$$E_S = \{E(s_i) = v; s_i \in S(G_1), v = [f_1^i, \dots, f_d^i] \in \mathbb{R}^d\} \text{ and } E_T = \{E(t_j) = v; t_j \in S(G_2), v = [f_1^j, \dots, f_d^j] \in \mathbb{R}^d\}$$

### 3.3. Building of Data

#### 3.3.1. Generation of Positive Links

From the sets of valid entity alignments, we generated the set of positive links (PL), representing pairs of aligned entities (see P3, Figure 1).  $v_{max}$  represents the size of the set  $PL$ . In mathematical notation we have :  $PL = \{(s_a, t_b, 1)_{ab}; 0 \leq ab < v_{max}, s_a \in S(V_{data}) \text{ and } t_b \in O(V_{data}), a = b\}$ .

#### 3.3.2. Building of Training and Testing Data

To train and test the models, we built datasets by concatenating the embeddings of positive entity pairs (vector addition operation) and any other concatenated pairs will be considered as negatives. (see P4, Figure 1).  $u_{max}$  represents the size of the set  $NL_{data}$ . In mathematical notation, we have:

$$PL_{data} = \{(s_a, t_b, E(s_a) + E(t_b), 1); s_a \in S(V_{data}) \text{ and } t_b \in O(V_{data}); a = b\}, NL_{data} = \{(s_k, t_l, E(s_k) + E(t_l), 0)_{kl}; s_k \in S(V_{data}) \text{ and } t_l \in O(V_{data}) \text{ and } (s_k, t_l, 1) \notin PL \text{ or } k! = l, 0 \leq kl < u_{max}\}$$

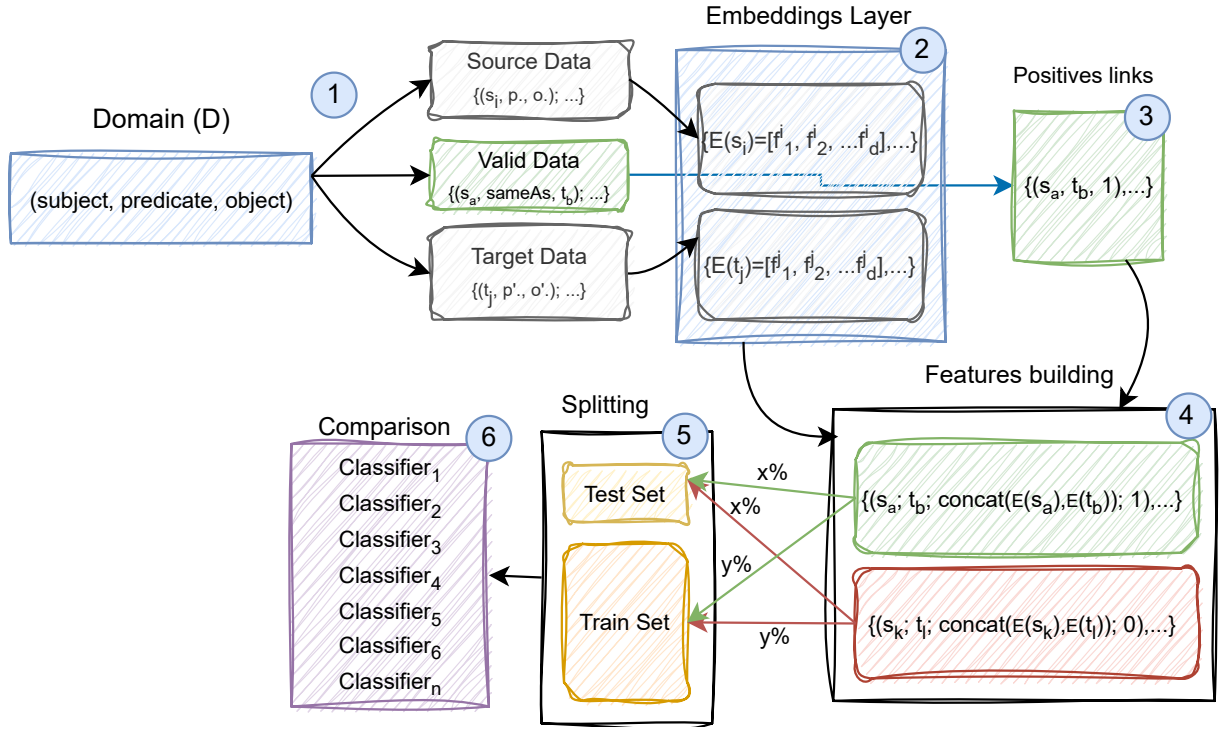


Fig. 1. Overview of the process.

### 3.3.3. Data Splitting

For training, we extracted a percentage ( $y\%$ ) from the previously calculated sets of positive ( $PL_{data}$ ) and negative ( $NL_{data}$ ) alignments, and for testing, a percentage ( $x\%$ ) (see P5, Figure 1).

### 3.4. Classifiers Evaluation

Finally, we evaluated the performance of our split data using different selected classifiers, measuring metrics such as accuracy, precision, recall, and F1-score (see P6, Figure 1).

#### 3.4.1. Accuracy

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \quad (1)$$

**Description:** Accuracy measures the proportion of correctly classified samples among all samples. It is a global metric of model performance.

#### 3.4.2. Precision

$$\text{Precision} = \frac{\text{Number of True Positives}}{\text{Number of True Positives} + \text{Number of False Positives}} \quad (2)$$

**Description:** Precision measures the proportion of correct positive predictions among all positive predictions. It indicates how precise the model is when predicting the positive class.

### 3.4.3. Recall

$$\text{Recall} = \frac{\text{Number of True Positives}}{\text{Number of True Positives} + \text{Number of False Negatives}} \quad (3)$$

**Description:** Recall measures the proportion of true positives among all truly positive examples. It indicates how well the model is at capturing all entities of the positive classes.

### 3.4.4. F1-score

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

**Description:** The F1-Score is a metric combining precision and recall into a single value. It is useful when you want to strike a balance between precision and recall, especially when dealing with imbalanced classes.

## 4. Experiments

Five evaluation datasets (Table 3), namely Anatomy, Doremus, Spimbench Small, Spimbench Large, and UOBM, sourced from the OAEI (Ontology Alignment Evaluation Initiative) benchmarks, have been employed in this study. These datasets have been utilized to assess the performance of various classification methods. To maintain computational efficiency, the dimensionality of the embedding vectors goes from 10 to 50 with 10 as the gap. The RDF2Vec algorithm employs default parameters, such as random walks of size 5, and sequences of length 1, to generate semantic representations of RDF data. In table 4, For each dimension, a standard split of 70% for training and 30% for testing has been applied to these datasets to ensure consistency in the evaluation process. To conduct experiments, nine classifiers were selected in Figure 2.

### 4.1. Datasets

Experiments have been evaluated on various datasets popular in the OAEI community. The OAEI community consistently offers challenges in entity alignment in the RDF format. We have curated a collection of datasets from IM@OAEI, spanning multiple editions. Among these datasets in Table 3, we have chosen to work with various sources, each bringing its unique complexity and specific characteristics.

- Anatomy<sup>1</sup>: The dataset associates anatomical terms of the adult mouse with those of the NCI Thesaurus to align the two ontologies despite linguistic and conceptual differences. It includes pairs of corresponding anatomical terms with metadata to facilitate alignment and interpretation of results.
- Doremus<sup>2</sup>: This dataset aims to discover the 250 existing alignments between instances from the Philharmonie de Paris (approximately 3,005 instances) and those from the "Bibliothèque nationale de France" (around 2,597 instances), making it a rich resource for cultural data.
- Spimbench (Small and Large 2019) [41]: They are described using a rich ontology with various OWL constructs. The small-sized datasets contain around 10,000 triples, with over 1,000 instances per file, and the large-sized consists of around 51,000 triples with more than 5,380 instances per entry.
- University Ontology Benchmark (UOBM): The University Ontology Benchmark (UOBM) is an extension of the Lehigh University Benchmark (LUBM) and is designed to evaluate the performance of ontology-based systems<sup>3</sup>. It includes a university domain ontology, customizable synthetic data, a set of test queries, and performance metrics<sup>4</sup>.

This diversity of datasets enables us to conduct varied evaluations and measure its performance in different contexts.

<sup>1</sup><https://oaei.ontologymatching.org/2020/anatomy/>

<sup>2</sup><https://github.com/DOREMUS-ANR/legato>

<sup>3</sup><https://swat.cse.lehigh.edu/projects/lubm/>

<sup>4</sup><https://www.cs.ox.ac.uk/isg/tools/UOBMGenerator/>

Table 3  
Dataset Statistics

Datasets	File	# Subjects	# Predicates	# of Relations (Facts)
Anatomy	source	8179	11	15958
	target	26144	11	35354
	valid	3006	1	1516
Doremus	source	5022	34	10432
	target	4045	38	8409
	valid	501	1	500
Spimbench Small	source	7263	66	10883
	target	7346	86	10868
	valid	525	1	598
Spimbench Large	source	29186	66	46223
	target	29186	89	46377
	valid	2646	1	3022
UOBM Small	source	5851	48	14625
	target	6648	211	14191
	valid	2355	1	2354

Table 4  
Train and Test Set Statistics

Datasets	Set	# Positives	# Negatives	Total
Anatomy	test	467	443	910
	train	1049	1073	2122
Doremus	test	75	68	143
	train	163	170	333
Spimbench Small	test	96	84	180
	train	203	215	418
Spimbench Large	test	466	441	907
	train	1045	1070	2115
UOBM Small	test	364	343	707
	train	813	834	1647

Orders	Names	Orders	Names
1	AdaBoostClassifier	6	Support Vector Classifier (SVC)
2	Random Forest Classifier	7	K-Nearest Neighbors (KNN)
3	XGBoostClassifier	8	Decision Tree Classifier
4	ExtraTreesClassifier	9	Gaussian Naive Bayes
5	Logistic Regression		

Fig. 2. Names of the classifiers with the associated value in the histogram.

#### 4.2. Comments on results

In Figures 3, 4, 5, and 6 below, the analysis of the results of entity alignment classification based on embeddings reveals several key observations. Firstly, it is evident that the classifiers used in this study do not guarantee significant performance (F1-score is lower than 0.7) on these extracted features. Evaluation metrics such as Accuracy, Precision, Recall, and F1-score show relatively modest values, indicating that entity alignment classification based on RDF embeddings is a complex challenge. Furthermore, it is interesting to note that metrics appear nearly identical for most classifiers, suggesting a sort of performance stagnation on these extracted features. This uniformity of performance may indicate that features extracted from RDF embeddings may not provide enough discrimina-

1 tive information for classifiers to make meaningful distinctions between classes. Another crucial observation is the 1  
2 similar metrics among classifiers because they do not truly excel at handling imbalanced data. In many real-world 2  
3 situations, entity alignment data can be imbalanced, meaning that some classes may be equally represented between 3  
4 them. Classifiers may struggle to handle this balance, leading to similar metrics. 4  
5

## 6 5. Discussion 6

7  
8  
9 The results obtained in this study raise several important questions and considerations regarding the classifica- 9  
10 tion of entity alignments based on RDF embeddings as seen in Figures 3, 4, 5, and 6. Firstly, it is evident that this 10  
11 classification task is complex and challenging, as evidenced by the modest performance of classifiers may be due 11  
12 to the under-performance of the embedding approach. Moreover, the mean value of the performances is around 12  
13 0.6, which translates into absolute non-confidence in the 9 best models evaluated. RDF embeddings, while rich 13  
14 in semantic information, may not always provide features that are sufficiently discriminative for this specific task. 14  
15 More sophisticated feature extraction techniques or RDF graph-specific pre-processing approaches may be needed 15  
16 to enhance performance. Furthermore, the similarity of metrics across different classifiers suggests that the choice 16  
17 of the classification model may not be the determining factor in this scenario. Instead, other factors such as the 17  
18 quality of training data, class balance, or the intrinsic nature of the classification task might play a significant role in 18  
19 performance. Handling imbalanced data is a major challenge in this context, as indicated by similar metrics. Classi- 19  
20 fiers may struggle to generalize effectively when some classes are overrepresented compared to others. Resampling 20  
21 techniques, class weighting, or other strategies may be necessary to mitigate this effect. Finally, it is important to 21  
22 note that RDF embeddings, while perhaps not ideally suited for entity alignment classification, have other valuable 22  
23 potential uses, including finding similar entities or discovering patterns in RDF graphs. In summary, these results 23  
24 highlight the complexity of entity alignment classification based on RDF embeddings and call for future research 24  
25 to explore more sophisticated approaches, address imbalanced data, and optimize classifier parameters to improve 25  
26 performance in this field. 26  
27

## 28 6. Conclusion 28

29  
30  
31 This study investigates the challenges of entity alignment classification using RDF embeddings. We leverage the 31  
32 OAEI community’s datasets, transforming each entity into feature vectors via the RDF2Vec embedding approach. 32  
33 Utilizing ground truth data, we establish a 70%/30% split for training and testing sets, respectively. We observe 33  
34 modest and similar performance across the classifiers by applying the nine best existing classifiers to the training data 34  
35 and evaluating the remaining 30%. Additionally, data imbalance issues become apparent. Despite these findings, 35  
36 RDF embeddings retain their value in other contexts. Future research should explore more sophisticated and robust 36  
37 classification approaches, alongside addressing data balance, to improve the efficiency of tasks like entity alignment 37  
38 classification. 38  
39

## 40 Acknowledgements 40

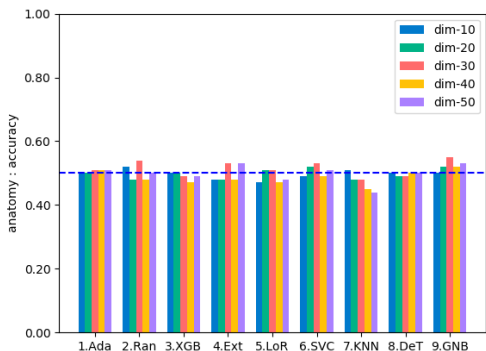
41  
42  
43 This work benefited from access to the high-performance computing resources of IDRIS under the 2024- 43  
44 AD010315119R1 allocation granted by GENCI and is supported by the French National Research Agency (ANR) 44  
45 within the framework of the DIG-AI-DL project, grant number ANR-22-CE23-0012. 45  
46

## 47 References 47

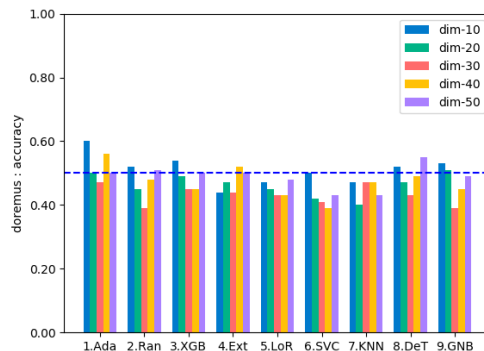
- 48  
49  
50 [1] Cox, D. Regression models and life-tables. *Journal Of The Royal Statistical Society: Series B (Methodological)*. **20**, 187-220 (1958) 49  
51 [2] Cortes, C. & Vapnik, V. Support-vector networks. *Machine Learning*. **20**, 273-297 (1995) 50  
[3] Quinlan, J. Induction of decision trees. *Machine Learning*. **1**, 81-106 (1986) 51

- [4] Breiman, L. Random forests. *Machine Learning*. **45**, 5-32 (2001)
- [5] Friedman, J. Greedy function approximation: A gradient boosting machine. *Annals Of Statistics*. pp. 1189-1232 (2001)
- [6] Freund, Y. & Schapire, R. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal Of Computer And System Sciences*. **55**, 119-139 (1997)
- [7] Lewis, D. Naive (Bayes) at forty: The independence assumption in information retrieval. *European Conference On Machine Learning*. pp. 4-15 (1998)
- [8] Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Transactions On Information Theory*. **13**, 21-27 (1967)
- [9] Bishop, C. *Neural networks for pattern recognition*. (Oxford university press,1995)
- [10] LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings Of The IEEE*. **86**, 2278-2324 (1998)
- [11] Ristoski, P. & Paulheim, H. Rdf2vec: RDF graph embeddings for data mining. *International Semantic Web Conference*. pp. 498-514 (2016)
- [12] Allen, T., Sherborne, T. & Eickhoff, C. ComplEx-ComplEx-KG: Learning Compositional Embeddings of Knowledge Graphs with Complementarity. *Proceedings Of The 15th Conference Of The European Chapter Of The Association For Computational Linguistics: Volume 2, Short Papers*. pp. 151-157 (2021)
- [13] Sun, Z., Deng, Z., Nie, J. & Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **33** pp. 9090-9097 (2019)
- [14] Lin, Y., Liu, Z., Sun, M., Liu, Y. & Zhu, X. TransD: Knowledge Graph Embedding via Adaptive Sparse Transfer Matrix. *Proceedings Of The 2015 Conference On Empirical Methods In Natural Language Processing*. pp. 1010-1019 (2015)
- [15] Ristoski, P., Rosati, J., Di Noia, T., De Leone, R. & Paulheim, H. RDF2Vec: RDF graph embeddings and their applications. *Semantic Web*. **10** pp. 1-32 (2018,8)
- [16] Lenat, D. & Feigenbaum, E. On the thresholds of knowledge. *Proceedings Of The International Workshop On Artificial Intelligence For Industrial Applications*. pp. 291-300 (1988)
- [17] Nezhadi, A., Shadgar, B. & Osareh, A. Ontology Alignment Using Machine Learning Techniques. *International Journal Of Computer Science & Information Technology (IJCSIT)*. **3** (2011,5)
- [18] Modarres, Z., Shabankhah, M. & Kamandi, A. Making AdaBoost Less Prone to Overfitting on Noisy Datasets. *2020 6th International Conference On Web Research (ICWR)*. pp. 251-259 (2020)
- [19] Kigo, S., Omondi, E. & Omolo, B. Assessing predictive performance of supervised machine learning algorithms for a diamond pricing model. *Scientific Reports*. **13**, 17315 (2023)
- [20] Nayyer, N., Javaid, N., Akbar, M., Aldegheishem, A., Alrajeh, N. & Jamil, M. A New Framework for Fraud Detection in Bitcoin Transactions through Ensemble Stacking Model in Smart Cities. *IEEE Access*. (2023)
- [21] Khan, A., BinZiad, A. & Subaïi, A. Boosting Algorithm Choice in Predictive Machine Learning Models for Fracturing Applications. *SPE Asia Pacific Oil And Gas Conference And Exhibition*. pp. D011S009R003 (2021)
- [22] Pagliaro, A. Forecasting Significant Stock Market Price Changes Using Machine Learning: Extra Trees Classifier Leads. *Electronics*. **12**, 4551 (2023)
- [23] Arathi, A., HariKrishna, M. & Mohan, M. Machine Learning-Based Gap Acceptance Model for Uncontrolled Intersections Under Mixed Traffic Conditions. *Conference Of Transportation Research Group Of India*. pp. 3-19 (2021)
- [24] Li, M., Liu, Y., Liu, X., Sun, Q., You, X., Yang, H., Luan, Z., Gan, L., Yang, G. & Qian, D. The deep learning compiler: A comprehensive survey. *IEEE Transactions On Parallel And Distributed Systems*. **32**, 708-727 (2020)
- [25] Cunningham, P. & Delany, S. k-Nearest neighbour classifiers: (with Python examples). *ArXiv Preprint ArXiv:2004.04523*. (2020)
- [26] Maimon, O. & Rokach, L. *Data mining with decision trees: theory and applications*. (World scientific,2014)
- [27] Zhang, M., Peña, J. & Robles, V. Feature selection for multi-label naive Bayes classification. *Information Sciences*. **179**, 3218-3229 (2009)
- [28] Touzani, S., Granderson, J. & Fernandes, S. Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy And Buildings*. **158** pp. 1533-1543 (2018)
- [29] Bujang, S., Selamat, A., Krejcar, O., Mohamed, F., Cheng, L., Chiu, P. & Fujita, H. Imbalanced Classification Methods for Student Grade Prediction: A Systematic Literature Review. *IEEE Access*. (2022)
- [30] Nezhadi, A. H., Shadgar, B., & Osareh, A. (2011). Ontology alignment using machine learning techniques. *International Journal of Computer Science & Information Technology*, 3(2), 139.
- [31] Nkisi-Orji, Ikechukwu, et al. "Ontology alignment based on word embedding and random forest classification." *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*. Springer International Publishing, 2019.
- [32] Bulygin, L. (2018, October). Combining lexical and semantic similarity measures with machine learning approach for ontology and schema matching problem. In *Proc. Int. Conf. Data Anal. Manage. Data Intensive Domains (DAMDID/RCDL)* (pp. 245-249).
- [33] Koppad, S., Basava, A., Nash, K., Gkoutos, G. V., & Acharjee, A. (2022). Machine learning-based identification of colon cancer candidate diagnostics genes. *Biology*, 11(3), 365.
- [34] Shvaiko, P., & Euzenat, J. (2011). Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1), 158-176.
- [35] Liu, L., Yang, F., Zhang, P., Wu, J. Y., & Hu, L. (2012). SVM-based ontology matching approach. *International Journal of Automation and Computing*, 9, 306-314.
- [36] Koech, G., & Fonou-Dombeu, J. V. (2021). K-nearest neighbors classification of semantic web ontologies. In *Model and Data Engineering: 10th International Conference, MEDI 2021, Tallinn, Estonia, June 21–23, 2021, Proceedings 10* (pp. 241-248). Springer International Publishing.

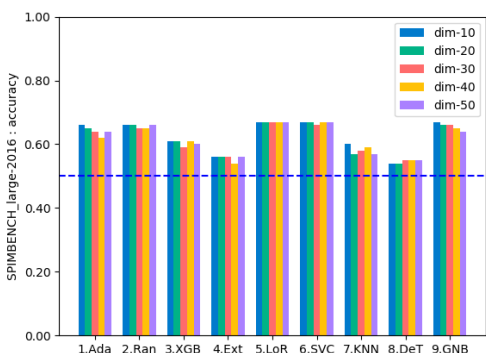
- [37] Amrouch, S., Mostefai, S., & Fahad, M. (2016). Decision trees in automatic ontology matching. *International Journal of Metadata, Semantics and Ontologies*, 11(3), 180-190.
- [38] Shadgara, B., Nejhadia, A. H., & Osareha, A. (2011). Ontology alignment using machine learning techniques. *International Journal of Computer Science and Information Technology*, 3.
- [39] Bulygin, L., & Stupnikov, S. A. (2019, October). Applying of Machine Learning Techniques to Combine String-based, Language-based and Structure-based Similarity Measures for Ontology Matching. In *DAMDID/RCDL* (pp. 129-147).
- [40] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- [41] Saveta, T., Daskalaki, E., Flouris, G., Fundulaki, I., Herschel, M., & Ngonga Ngomo, A.-C. (2015). Pushing the limits of instance matching systems: A semantics-aware benchmark for linked data. *Proceedings of the 24th International Conference on World Wide Web*, 105–106.
- [42] Justo-Silva, R., Ferreira, A., & Flintsch, G. (2021). Review on machine learning techniques for developing pavement performance prediction models. *Sustainability*, 13(9), 5248.



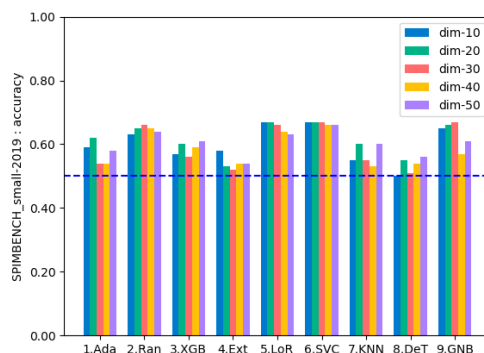
(a) Classifiers Accuracy on Five Different Embedding Dimensions for the Anatomy Dataset.



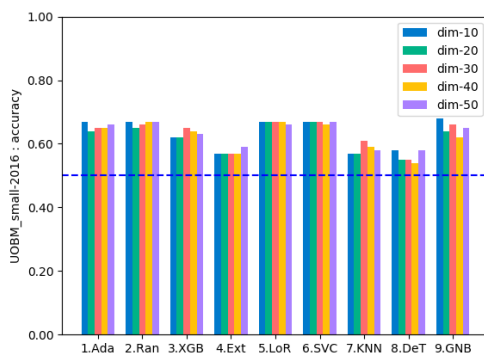
(b) Classifiers Accuracy on Five Different Embedding Dimensions for the Doremus Dataset.



(c) Classifiers Accuracy on Five Different Embedding Dimensions for the Spimbench large Dataset.

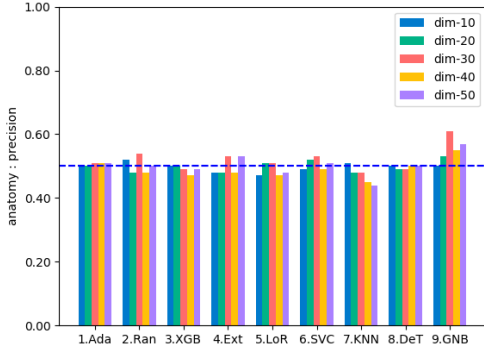


(d) Classifiers Accuracy on Five Different Embedding Dimensions for the Spimbench small Dataset.

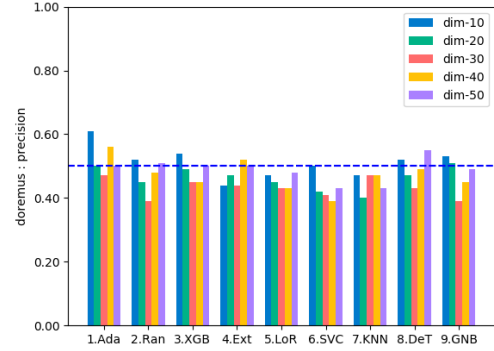


(e) Classifiers Accuracy on Five Different Embedding Dimensions for the UOBM small Dataset.

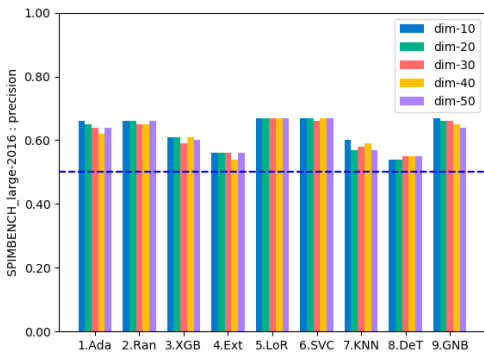
Fig. 3. Classifiers Accuracy on Five Different Embedding Dimensions for all Datasets.



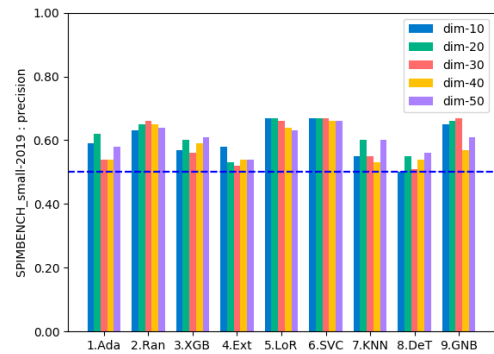
(a) Classifiers Precision on Five Different Embedding Dimensions for the Anatomy Dataset.



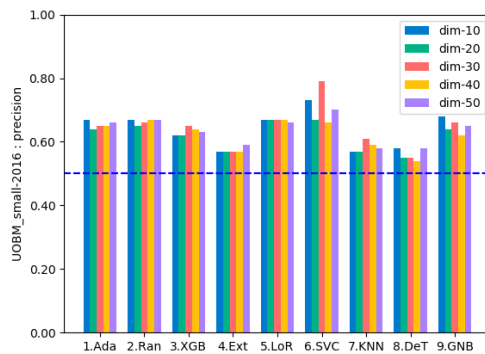
(b) Classifiers Precision on Five Different Embedding Dimensions for the Doremus Dataset.



(c) Classifiers Precision on Five Different Embedding Dimensions for the Spimbench large Dataset.

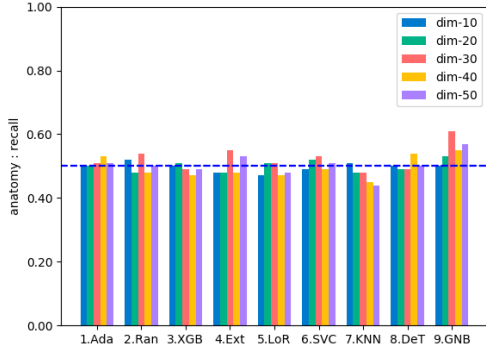


(d) Classifiers Precision on Five Different Embedding Dimensions for the Spimbench small Dataset.

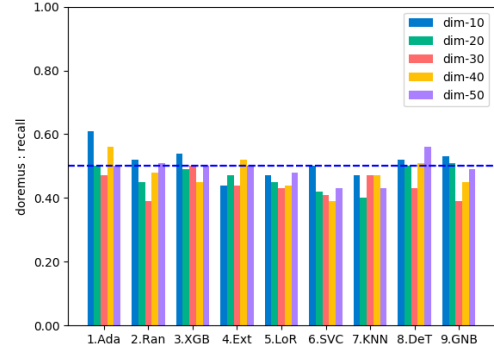


(e) Classifiers Precision on Five Different Embedding Dimensions for the UOBM small Dataset.

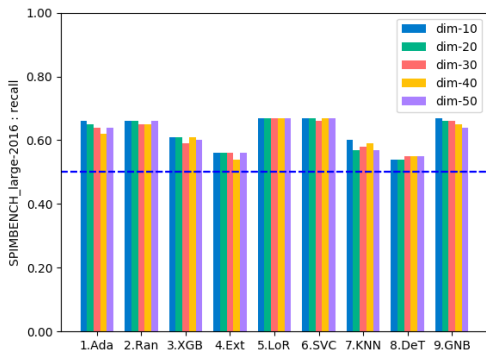
Fig. 4. Classifiers Precision on Five Different Embedding Dimensions for all Datasets.



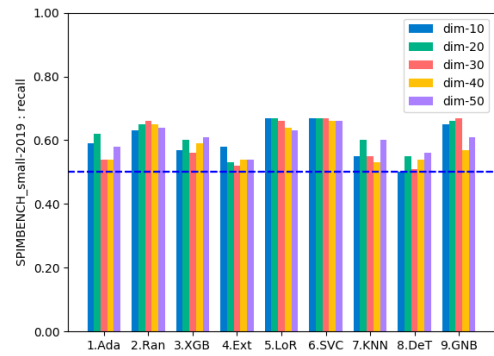
(a) Classifiers Recall on Five Different Embedding Dimensions for the Anatomy Dataset.



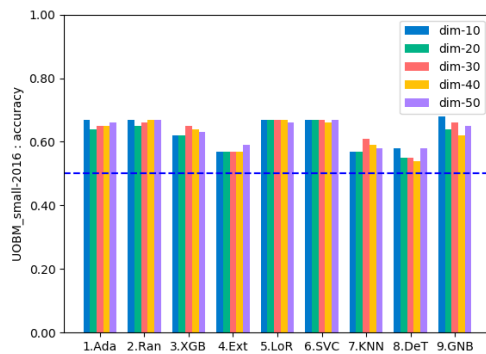
(b) Classifiers Recall on Five Different Embedding Dimensions for the Doremus Dataset.



(c) Classifiers Recall on Five Different Embedding Dimensions for the Spimbench large Dataset.

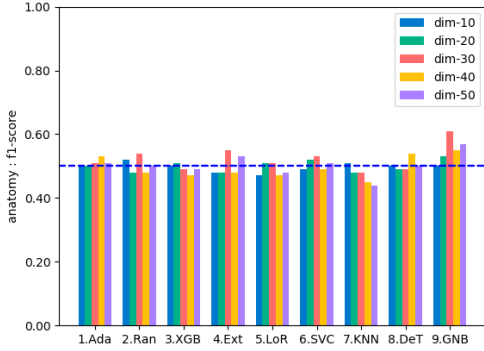


(d) Classifiers Recall on Five Different Embedding Dimensions for the Spimbench small Dataset.

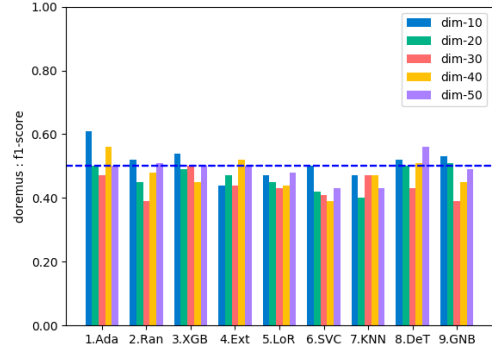


(e) Classifiers Recall on Five Different Embedding Dimensions for the UOBM small Dataset.

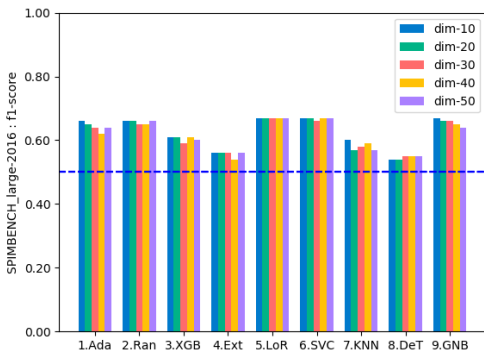
Fig. 5. Classifiers Recall on Five Different Embedding Dimensions for all Datasets.



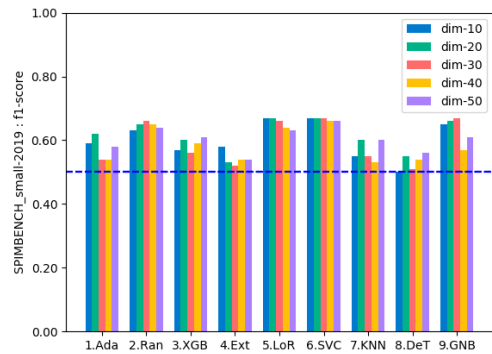
(a) Classifiers F1-score on Five Different Embedding Dimensions for the Anatomy Dataset.



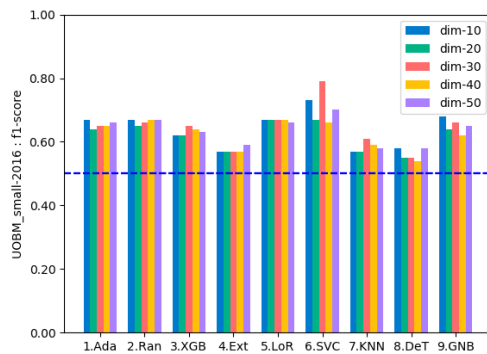
(b) Classifiers F1-score on Five Different Embedding Dimensions for the Doremus Dataset.



(c) Classifiers F1-score on Five Different Embedding Dimensions for the Spimbench large Dataset.



(d) Classifiers F1-score on Five Different Embedding Dimensions for the Spimbench small Dataset.



(e) Classifiers F1-score on Five Different Embedding Dimensions for the UOBM small Dataset.

Fig. 6. Classifiers F1-score on Five Different Embedding Dimensions for all Datasets.