The Impact of Imbalanced training Data on Local matching learning of ontologie

Amir Laadhar, Faiza Ghozzi, Imen Megdiche, Franck Ravat, Olivier Teste, Faiez Gargouri

The Impact of Imbalanced Training Data on Local Matching Learning of Ontologies

Amir Laadhar[1], Faiza Ghozzi[2], Imen Megdiche[1], Franck Ravat[1], Olivier Teste[1], and Faiez Gargouri[2]

[1] Institut de Recherche en Informatique de Toulouse, Toulouse, France
[2] MIRACL, Sfax University, Sakiet Ezzit, Sfax, Tunisie

Abstract. Matching learning corresponds to the combination of ontology matching and machine learning techniques. This strategy has gained increasing attention in recent years. However, state-of-the-art approaches implementing matching learning strategies are not well-tailored to deal with imbalanced training sets. In this paper, we address the problem of the imbalanced training sets and their impacts on the performance of the matching learning in the context of aligning biomedical ontologies. Our approach is applied to local matching learning, which is a technique used to divide a large ontology matching task into a set of distinct local sub-matching tasks. A local matching task is based on a local classifier built using its balanced local training set. Thus, local classifiers discover the alignment of the local sub-matching tasks. To validate our approach, we propose an experimental study to analyze the impact of applying conventional resampling techniques on the quality of the local matching learning.

Keywords: Imbalanced Training Data · Machine Learning · Ontology Matching · Semantic Web

1 Introduction

Biomedical ontologies such as SNOMED CT, the National Cancer Institute Thesaurus (NCI), and the Foundational Model of Anatomy (FMA) are used in the biomedical engineering systems [23]. These ontologies are developed based on different modeling views and vocabularies. The integration of these knowledge graphs requires efficient biomedical ontology matching systems [22, 7]. The mapping between heterogeneous ontologies enables data interoperability.

Ontology mapping becomes a challenging and time-consuming task due to the size and the heterogeneity of biomedical ontologies. In this domain, Faria et al. [8] identified different challenges such as handling large ontologies or exploiting background knowledge. In addition to these problems, we highlight the importance of ensuring good matching quality while aligning large ontologies. Among the cited strategies to deal with large ontologies, we find search-space reduction techniques encompassing two sub-strategies: partitioning and pruning [7]. The partitioning approach divides a large matching task into a set of smaller matching tasks, called partitions or blocks [23]. Each partition focuses

on a specific context of the input ontologies. The ontology alignment process consists of aligning similar partition-pairs. The partitioning process aims to decrease the matching complexity of a large matching task. A line of work performs the partitioning process of pairwise ontologies to align the set of identified similar partitions (e.g., [9, 1, 26, 4, 11]). The state-of-the-art partitioning approaches use global matching settings (e.g., matchers choice, thresholds and weights) for all extracted partition-pairs [23]. Therefore, they do not employ any local tuning applied to each partition-pair to maximize the matching quality. Despite the existing work applying global matching process, we perform a local matching process over each extracted partition-pair. The local matching resolved by machine learning techniques, known as "local matching learning", requires a set of local training sets. Imbalanced training sets are one of the main issues occurring while dealing with ontology matching learning. Imbalanced data typically refers to a classification problem where the number of observations per class is not equally distributed [18]. Current matching learning work does not consider the resampling process to resolve the problem of imbalanced matching learning training data. Nonetheless, resampling is essential to deliver better matching learning accuracy.

In this paper, the main novelty is studying the impact of the imbalanced training data issue in the case of aligning large biomedical ontologies. Unlike state-of-the-art approaches, we automatically derive a local training set for each local matching learning classifier based on external biomedical knowledge resources. We automatically generate a local training set for each classifier without the use of any gold standard. Furthermore, we employ existing resampling methods to balance local training sets of local classifiers, and to align each partition-pair. Then, we evaluate the matching accuracy after applying different resampling techniques on the local matching tasks. To the best of our knowledge, there is a lack of works that automatically generate and resample matching learning training data. In sum, the contributions of this paper are the following:

- Local matching learning of biomedical ontologies;
- Automatically generating labeled local training sets, which are inferred from external biomedical knowledge bases to build local-based classifiers;
- Comparative study of the resampling methods to balance the generated local training sets.

The remainder of this paper is organized as follows: the next section presents the related work. Section 3 introduces preliminaries for the local matching approach and the local training set. Section 4 presents an overview of the proposed local matching architecture. In Section 5, we present our method for local matching learning. In Section 6, we perform a comparative study of the state-of-the-art resampling techniques in order to resample imbalanced local training sets. Finally, Section 7 concludes this paper and gives some perspectives.

2 Related Work

Faria et al. [8] identified the different challenges to align large biomedical ontologies. Ensuring good quality alignments while aligning these ontologies is

challenging. To cope with these issues, we propose to employ matching learning techniques in order to fully automate the ontology matching process. Therefore, matching learning automates the alignment process while ensuring quality alignment independently from the matching context. In this section, we review the state-of-the-art matching learning strategies as well as the resampling techniques. Resampling methods could be applied to balance imbalanced training sets.

2.1 Ontology Matching Learning

There has been some relevant work dealing with supervised matching learning [10, 19, 20, 6]. Machine learning approaches for ontology alignment usually follow two phases [7]:

1. In the training phase, the machine learning algorithm learns the matching settings from a training set. This training set is usually created from the reference alignments of the same matching task.
2. In the classification phase, the generated classifier is applied over the input ontologies to classify the candidate alignments of the input ontologies.

Eckert et al. [6] built a meta-learner strategy to combine multiple learners. Malform-SVM [10] constructed a matching learning classifier from the reference alignments through a set of element level and structural level features. Nezhadi et al. [19] presented a machine learning approach to aggregate different types of string similarity measures. The latter approach is evaluated through a relatively small bibliographic matching track provided by the OAEI benchmark. Yam++ [20] defined a decision tree classifier based on a training set with different similarity measures. The decision tree classifier is built from the reference alignments and evaluated through the matching tasks. Nkisi-Orji et al. [21] proposed a matching learning classifier based on 23 features. Wang et al. [16], feed a neural network classifier with 32 features covering commonly used measures in the literature. A global classifier is built based on all the 32 features. Both approaches [21, 16] do not mention if the training set is balanced or not. Moreover, the training set is generated from the reference alignments.

Existing matching learning approaches build their machine learning classifiers from the reference alignments or derive it manually from a particular matching task [7]. We automatically generate a local training set for each sub-matching task. We do not use any reference alignments or user interactions to build the local training sets. Each local machine learning classifier is based on its local training set, which provides adequate matching settings for each sub-matching context.

2.2 Training Set Resampling

Classification problems often suffer from data imbalance across classes. This is the case when the size of instances from one class is significantly higher or lower relative to the other classes. A small difference often does not matter [18]. However, if there is a modest class imbalance in training data like 4:1, it can

cause misleading classification accuracy. Imbalanced data refers to classification problems where we have unequal instances for different classes.

Most of machine learning classification algorithms are sensitive to imbalanced training data. An imbalanced training data will bias the prediction classifier towards the more common class. This happens because machine learning algorithms are usually designed to improve accuracy by reducing the error. Thus, they do not take into account the class distribution of classes. Different methods have been proposed in the state-of-the-art for handling the data imbalance [5]. The common approaches [3] to generate a balanced dataset from an imbalanced one are undersampling, oversampling, and their combination:

- Undersampling approach balances the dataset by reducing the size of the abundant class by keeping all samples in the rare class and randomly selecting an equal number of samples in the abundant class;
- Oversampling approach is used when the quantity of data is insufficient. It tries to balance dataset by increasing the size of rare samples. Rather than removing of abundant samples, new rare samples are generated;
- Performing a combination of oversampling and undersampling can yield better results than either in isolation.

Most of the state-of-the-art matching learning approaches neglect the problem of the imbalanced dataset. Existing work does not give any importance to resampling. However, the resampling method can strongly affect the obtained accuracy by the matching learning strategy. In this paper, we propose to resample training data in the context of local matching learning of biomedical ontologies. We apply the state-of-the-art resampling techniques on the local training data. Then, we study the impact of the applied resampling techniques on the local matching accuracy.

3 Preliminaries

In this section, we briefly present the fundamental definitions used in our work.

Definition 1 (Ontology partition).
An ontology partition $p_{i,k}$ *of an ontology* $\mathcal{O}_i$ *is a sub-ontology denoted by* $p_{i,k} = (\mathcal{V}_{i,k},\mathcal{E}_{i,k})$*, such as :*

- $\mathcal{V}_{i,k} = \{e_{i,k,1},...,e_{i,k,m_k}\}$, $\mathcal{V}_{i,k} \subseteq \mathcal{V}_i$*. is a finite set of classes of the ontology partition,*
- $\mathcal{E}_{i,k} = \{(e_{i,k,x},e_{i,k,y}) \mid (e_{i,k,x},e_{i,k,y}) \in \mathcal{V}_{i,k} \ \exists \ (e_{i,k,x},e_{i,k,y}) \in \mathcal{E}_{i,k} \}$, $\mathcal{E}_{i,k} \subseteq \mathcal{E}_i$*. is a finite set of edges, where an edge encodes the relationship between two classes into an ontology partition*

Definition 2 (Set of ontology partitions).
An ontology $\mathcal{O}_i$ *can be divided into a set of ontology partitions* $\mathcal{P}_i = \{p_{i,1},..,p_{i,s_i}\}$.

- $\forall k \in [1..s_i], \mathcal{V}_{i,k} \neq \emptyset$;
- $\bigcup_{k=1}^{s_i} \mathcal{V}_{i,k} = \mathcal{V}_i$;
- $\forall\ k \in [1..s_i]$, $\forall\ l \in [1..s_i]$, $k \neq l$, $\mathcal{V}_{i,k} \cap \mathcal{V}_{i,l} = \emptyset$.

Definition 3 (Local-based training Set).
For each local matching $lm_{ij,q}$ of $\mathcal{LM}_{ij}$, we automatically generate a local training set denoted $ts_{ij,q}$. A local training set $ts_{ij,q}$ for a local matching task $lm_{ij,q}$ of $\mathcal{LM}_{ij}$ is denoted by $ts_{ij,q} = \{f_{i_1}, ..., f_{n_{i_q}}\}$. Each local training set $ts_{ij,q}$ contains a set of features associated with a prediction class attribute (match/not match). We denote $ts_{ij,q} = \{f_{ijq,1}, ..., f_{ijq,r}, c_{ijq}\}$. Since we are dealing with a binary classification task, $c_{ijq} \in \{0,1\}$. A set of local machine learning classifiers is generated from a set of local training sets $\mathcal{T}s_{ij}$, denoted $\mathcal{T}s_{ij,q} = \{ts_{ij,1}, ..., ts_{ij,q}\}$.

Hypothesis 1
For a local matching $\mathcal{LM}_{ij}$ between two ontologies $\mathcal{O}_i$ and $\mathcal{O}_j$ associated with a set of imbalanced local training sets $\mathcal{T}s_{ij} = \{ts_{ij,1}, ..., ts_{ij,q}\}$, performing the adequate resampling technique improves the accuracy of the local matching learning.

4 Local Matching Learning Architecture Overview

In Figure 1, we depict an architectural overview of the local matching workflow. This architecture follows three modules: (i) ontology indexing and partitioning, (ii) local matching learning and (iii) alignment evaluation. We participated in the Ontology Alignment Evaluation Initiative (OAEI)[3] of 2018 using this architecture [14]. In the following, we describe the different modules of our proposal.

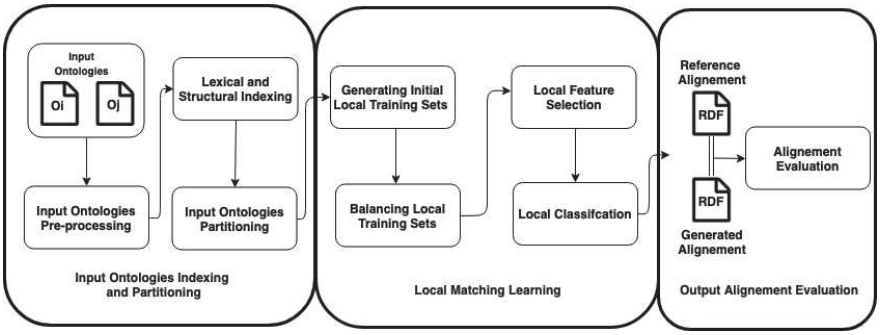

Fig. 1. Local matching architecture overview

Input Ontologies Indexing and Partitioning The input ontologies indexing is composed of two sequential steps: input ontologies pre-processing, and lexical and structural indexing. In the first step, we pre-process the lexical annotations. Thus, we apply the Porter stemming [15] and a stop word removal process over the extracted lexical annotations. In the second step, those lexical annotations are indexed. Moreover, we use structural indexing to store all the relationships between entities.

We then perform the partitioning of the input ontologies based on the approach of [13]. This partitioning approach is based on the Hierarchical Agglomerative Clustering (HAC) [17] to produce a set of partition-pairs with a sufficient coverage ratio and without producing any isolated partitions. Partitions with only one entity are considered as isolated. This partitioning process follows these steps:

[3] http://oaei.ontologymatching.org/2018/

1. Ontology partitioning processing: the HAC algorithm generates a list of structural similarity scores between all the entities of each input ontology. The structural similarity measure computes the structural relatedness between every pair of entities of one ontology.
2. Dendrogram construction: the HAC approach receives as an input the list of structural similarities of every input ontology. The HAC generates a dendrogram for every ontology. A dendrogram represents the structural representation of an input ontology.
3. Dendrogram cut: we cut the two generated dendrograms. A single cut of a dendrogram can result in a set of large partitions. To cope with this issue, we perform an automated multi-cut strategy of every resulted dendrogram. The multi-cut strategy results in a set of partitions for each ontology.
4. Finding similar partition-pairs: we find the set of similar partition-pairs between the two ontologies. These partition-pairs represent the set of local matching tasks, which will be employed by the next module.

The employed partitioning approach to generate local matching tasks is explained in depth in our previous research work [13].

Local Matching Learning The local matching approach is based on generating a local classifier for each sub-matching task. The local classifiers are generated based on a set of local training sets composed of element and structural level features. Each local classifier is based on adequate features to align its sub-matching task. These features are automatically selected based on a feature selection. The local matching approach is composed of the following steps:

1. Generating Initial Local Training Set: Initial local training set are generated for each local matching task. These local training sets are not balanced.
2. Resampling of the Local Training Data: During this step, local training sets are balanced using conventional resampling techniques. Resampling aims to balance local training sets for better classification accuracy.
3. Wrapper Local Feature Selection: We apply wrapper feature selection over each resampled local training set. This local feature selection aims to choose the adequate features for each local matching task.
4. Local Classification: Local classification aims to classify the candidate correspondences of a local matching task to be aligned or not. This process is based on a set of local classifiers generated from the set of local training sets.

We will provide an in-depth description of these steps in the next section.

Output Alignment Evaluation The generated output correspondences for every local matching task $lm_{ij,q}$ are unified to generate the final alignment file for the whole ontology matching task. The alignment file is compared to the reference alignment to evaluate the overall local matching $\mathcal{LM}_{ij}$ accuracy.

5 Local Matching Learning

In this section, we present the different steps of the local matching learning module depicted in the architecture of Figure 1.

Generating Initial Local Training Set: Each local matching task has its own specific context. Therefore, a local matching task should be aligned based on its adequate matching settings, such as the weight of each matcher and its threshold. To cope with this issue, a local based classifier should be built for each local matching task. Therefore, we automatically construct a supervised training set $ts_{ij,q}$ for each local matching task $lm_{ij,q}$. These training sets serve as the input for each local classifier. Labeled data for the class attribute c_{ijq} are usually hard to acquire. Existing work construct the labeled data either from the reference alignments or by creating it manually [20]. However, the reference alignments commonly do not exist. We automatically generate the local training sets without any user manual involvement or any reference alignments. We derive the positive mappings samples (minority class) of the class attribute c_{ijq} by combining the results of two methods: cross-searching and cross-referencing. This combination allows the enrichment of the local training sets in order to cover a wide range of biomedical ontologies.

For a given local matching $lm_{ij,q}$, we generate:

- Positive samples $ps_{ij,q} = \{(e_i, q, x, e_{j,q,y})\}$, the total number of these positive samples is $\mathcal{N} = |ps_{ij,q}|$.
- Negative samples $ns_{ij,q} = \{(\mathcal{V}_{i,q} \times \mathcal{V}_{j,q}) \setminus ps_{ij,q}\}$. Therefore, the total number of negative samples is $\mathcal{M} = \mathcal{N}(\mathcal{N}\text{-}1)$.

In the following, we present the cross-searching and the cross-referencing methods:

Cross-searching: Cross-searching employs external biomedical knowledge sources as a mediator between local matching tasks in order to extract bridge alignments. A bridge alignment is extracted if a similar annotation is detected between an entity of the bridge ontology and two entities of a local matching task. We consider bridge alignments as the positive samples. For example, in Figure 2,we extracted the following positive samples $PS_{ij,q} = \{(\mathrm{e}_1\mathcal{P}_q, \mathrm{e}_1\mathcal{P}_r), (\mathrm{e}_8\mathcal{P}_q, \mathrm{e}_2\mathcal{P}_r), (\mathrm{e}_5\mathcal{P}_q, \mathrm{e}_3\mathcal{P}_r)\}$, with $\mathcal{N}= 3$. These labeled classes are generated by cross-searching the two partitions and an external biomedical knowledge base. Therefore, we deduce that the number of negative samples $\mathcal{M} = \mathcal{N}\ (\mathcal{N}\text{-}1) = 6$ for the partition-pair $\mathcal{P}_q$ and $\mathcal{P}_r$ respectively from $\mathcal{O}_i$ and $\mathcal{O}_j$. *Cross-referencing*: we employ Uberon as an external biomedical knowledge source in order to derive positive samples for each local training set. Uberon is an integrated cross-species ontology covering anatomical structures and includes relationships to taxon-specific anatomical ontologies. Indeed, we explored the property "hasDbXref", which is mentioned in almost every class of Uberon. This property references the classes URI of external biomedical ontologies. We align every two entities of a given local matching task in case if one of their entities are both referenced in a single entity of Uberon. For example, the UBERON ontology includes references to different biomedical ontologies (via annotation property "hasDbXRef "). For instance, the class UBERON_0001275 ("pubis") of Uberon references the FMA class 16595 ("pubis") and NCI class C33423 ("pubic bone"). Therefore, the later entities construct a positive sample of a local training set [8].

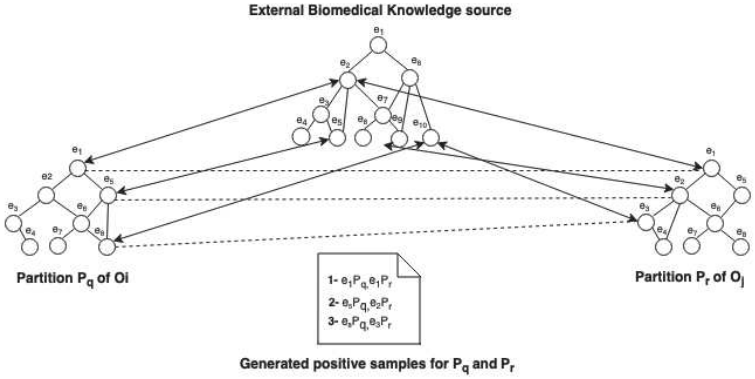

Fig. 2. Local training set extraction

Resampling of the Local Training Data: The training set is not balanced since the number of the negative samples $\mathcal{M}$ is higher than the number of positive samples $\mathcal{N}$. Therefore, we initially undersample each local training set $ts_{ij,q}$ by a heuristic method which consists of removing all the negative samples (majority class) having at least one element level feature equal to zero. The result of this initial treatment is not enough to balance the local training data, due to the high number of negative samples $\mathcal{M}$ (majority class) compared to the number of positive samples $\mathcal{N}$ (minority class). Hence, an additional sampling method is required to result in a balanced training set, we employ the state-of-the-art resampling methods to perform the undersampling of the majority class $\mathcal{M}$, oversampling the minority class $\mathcal{N}$ or combining both of the later technique. In section 5, we conduct a comparative study of applying resampling methods over imbalanced training data. Therefore, we obtain a balanced training set ($\mathcal{N}$=$\mathcal{M}$). We Denote by $r = \|\mathcal{N}\|/\|\mathcal{M}\|$ the ratio of the size of the minority class to the majority class.

The output of this step is a balanced local training set $ts_{ij,q}$ for each local matching task $lm_{ij,q}$. For instance, if a local matching process $\mathcal{LM}_{ij}$ is composed of three local matching tasks: $lm_{ij,1}$ $lm_{ij,2}$ and $lm_{ij,3}$, we respectively result in three local training sets $ts_{ij,1}$ $ts_{ij,2}$ $ts_{ij,3}$.

Wrapper Local Feature Selection The local matching $\mathcal{LM}_{ij}$ approach splits a large ontology matching problem into a set of smaller local matching tasks $lm_{ij,q}$. Each local matching task focuses on a specific sub-topic of interest. Therefore, it should be aligned based on its suitable features. We employ wrapper feature selection approaches in order to determine the suitable features for each local matching task $lm_{ij,q}$ among 23 structural-level and element-level features. Element level features consider intrinsic features of entities such as their textual annotations. Element level features refer to well-known similarity measure matchers, which can be classified into four groups: edit-distance, character-based, term-based and subsequence-based [13]. Structure level features consider the ontological neighborhood of entities in order to determine their similarity. We previously introduced all these features in a previous research work [13]. Feature

selection is performed over each local training set $ts_{ij,q}$ in order to build local classifiers. The later identifies the local alignments of a local matching task $lm_{ij,q}$. For example for a given local training sets: $ts_{ij,1}$, $ts_{ij,2}$ and $ts_{ij,3}$, we separately perform the feature selection over these three local training sets.

Local Classification Candidate correspondences of each local matching task $lm_{ij,q}$ are determined through performing the Cartesian product between the entities $\mathcal{V}_{i,q}$ and $\mathcal{V}_{j,q}$ of $lm_{ij,q}$. A local classifier classifies each candidate correspondence into a true or a false alignment. We build local classifiers using the Random Forest algorithm. We have selected Random Forest after comparing the efficiency of several machine learning algorithms for ontology matching learning [13].

6 Evaluation and Comparative Study

In this section, we evaluate the Hypothesis 1 "Resampling improves the accuracy of the local matching learning" by conducting a set of experiments. All experiments have been implemented in Java (weka library for classification) and Python (imblearn library for training sets resampling) on a MacOs operating system with 2.8 GHz Intel I7-7700HQ (4 cores) and 16 GB of internal memory. In the following subsections, we compare the accuracy of the commonly employed resampling methods (undersampling, oversampling and their combination) then we discuss the results. Our experiments are performed based on the dataset of the Evaluation Initiative of 2018, in particular, the ontology matching track of Anatomy.

6.1 Impact of Undersampling on Local Training Data

We evaluate four different methods of the widely employed undersampling method in order to balance the class distribution of the local training data. These methods are described as follows:

- **Random undersampling** is a non-heuristic method that aims to balance the class distribution through the random elimination of instances belonging to the majority class.
- **Tomek links** [24] removes unwanted overlap between classes where majority class links are removed until all minimally distanced nearest neighbor pairs are on the same class.
- **One-sided selection (OSS)** [12] aims at creating a training dataset composed only by "safe instances". This technique removes instances that are noisy, redundant, or near to the decision border. Similar to the other undersampling techniques, OSS removes only instances from the majority class.
- **Edited Nearest Neighbors** [25] method removes the instances of the majority class with prediction made by the K-means method is different from the majority class. Therefore, if an instance has more neighbors of a different class, this instance will be removed.

In Table 1, we depict the results of each undersampling method in terms of the obtained accuracy. The Edited Nearest Neighbors resulted in the highest F-Measure of 85.3%.

Table 1: Local Matching accuracy for each undersampling method

Undersampling method	Precision	Recall	F-Measure
Random Undersampling	65.9%	86.4%	74.8%
Tomek links	93.7%	77.4%	84.8%
One-sided selection	93.7%	77.1%	84.6%
Edited Nearest Neighbors	93.4%	78.4%	85.3%

6.2 Impact of Oversampling on Local Training Data

In this section, we perform the oversampling of the minority class instead of performing the undersampling of the majority class. There are several oversampling methods used in typical classification problems. The most common techniques are SMOTE [2] (Synthetic Minority Oversampling Technique) and random oversampling method:

- **SMOTE** oversamples the minority class by taking each positive instance and generating synthetic instances along a line segments joining their k nearest neighbors
- **Random oversampling** is a non-heuristic method that aims to balance class distribution through the random elimination of instances belonging to the minority class.

In the following Table 2, we depict the results of the local matching $\mathcal{LM}_{ij}$ after performing the oversampling of the local training sets for each local matching task $lm_{ij,q}$ of $\mathcal{LM}_{ij}$. We show below the results of applying SMOTE with $k = 5$ as the number of nearest neighbors in order to achieve a ratio $r = 1$. We also perform the random oversampling with a ratio r= 1We deduce from Table 2 that SMOTE outperforms the random oversampling method in terms of precision and F-Measure. We argue this result due to the randomly generated instances by the random oversampling method.

Table 2: Local Matching accuracy for each oversampling method

Oversampling method	Precision	Recall	F-Measure
Random oversampling	70.8%	82.8%	76.3%
SMOTE	86.7%	78.8%	82.6%

6.3 Combination of oversampling and undersampling on Local Training Data

It is possible to combine oversampling and undersampling techniques into a hybrid strategy. Common state-of-the-art methods [18, 3] include the combination of SMOTE and Tomek links, SMOTE and Edited Nearest Neighbors (ENN) or SMOTE and Random Undersampling. SMOTE is employed for the oversampling with a ratio r= 0.5 and each of the latter techniques is employed for undersampling. We evaluate these three methods by resampling the local training sets, then we perform the local matching process based on the generated local classifiers. In the following Table 3, we depict the results of each combination method. We deduce that the combination of SMOTE and Tomek Link results in the best accuracy in terms of F-Measure and Precision. The combination of SMOTE and Random Under sampling results in the best recall value due to the random nature of this approach. Therefore, it returns the highest number of alignments with the lowest precision compared to the other combination methods.

Table 3: Combining Oversampling and Undersampling techniques

Hybrid method	Precision	Recall	F-Measure
SMOTE + Random Undersampling	87.8%	81.3%	84.4%
SMOTE + ENN	88.4%	80.5%	84.3%
SMOTE + Tomek Link	92.0%	79.0%	85.0%

6.4 Discussion

According to the achieved results, we highlight the following points:

- The random undersampling and oversampling methods are unstable. A more deep study on the convergence of these methods can be investigated to argue their usage.
- We can observe in table 3 that combining SMOTE with undersampling methods decreases the matching accuracy.

We deduce that for the matching learning context, undersampling methods outperform the other resampling methods. We argue this result since the undersampling method removes redundant instances rather than creating new synthetic instances like the oversampling of the minority class. We conclude that the undersampling using ENN [25] yields to the best Precision and F-Measure. ENN removes instances of the majority class with prediction made by K-means method which are different from the majority class. We mention that we combined the use of cross-searching and cross-referencing with external resources in order to construct local training sets. The impact of each method on the resulted matching quality can be investigated. We tend to validate the hypothesis that the quality could depend on the use of methods used in the construction of the training data sets.

7 Conclusion

Ontology matching based on machine learning techniques has been an active research topic during recent years. In this paper, we focus on the problem of imbalanced learning training sets in the case of Local Matching Learning. To the best of our knowledge, there is a lack of works that automatically generate and resample local matching learning training sets. To perform the resampling of the local training sets, we evaluate the common state-of-the-art resampling techniques in order to improve the classification performance by employing the best technique. Our comparative study shows that the undersampling methods outperform the oversampling methods and their combination.

In future work, we tend to automate the choice of the resampling method for each local matching task. We will also evaluate the impact of the external knowledge resource on the global quality of our output alignments. Finally, we plan to experiment on different domain-based ontologies.

References

1. Algergawy, Alsayed, et al. "Seecont: A new seeding-based clustering approach for ontology matching." ADBIS. Springer, Cham (2015)
2. Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002)

3. Chawla, Nitesh V. "Data mining for imbalanced datasets: An overview." Data mining and knowledge discovery handbook. Springer, Boston, MA, (2009)
4. Chiatti, Agnese, et al. "Reducing the search space in ontology alignment using clustering techniques and topic identification." ICKC, ACM, (2015)
5. de Souto et al. "An empirical analysis of under-sampling techniques to balance a protein structural class dataset." ICNIP, Springer, Berlin, Heidelberg, (2006)
6. Eckert, Kai, Christian M, and Heiner S. "Improving ontology matching using meta-level learning." European Semantic Web Conference. Springer (2009)
7. Euzenat, J., Shvaiko, P. "Ontology matching, vol. 1." (2007)
8. Faria, D., Pesquita, C., Mott, I., Martins, C., Couto, F. M., Cruz, I. F. (2018). Tackling the challenges of matching biomedical ontologies. JBS, 9(1), 4.
9. Hu, Wei, Yuzhong Qu, and Gong Cheng. "Matching large ontologies: A divide-and-conquer approach." DKE V 67.1 (2008)
10. Ichise, Ryutaro. "Machine learning approach for ontology mapping using multiple concept similarity measures." 7th IEEE/ACIS, (2008).
11. Jiménez-Ruiz, Ernesto, et al. "We Divide, You Conquer: From Large-scale Ontology Alignment to Manageable Subtasks." Ontology Matching (2018)
12. Kubat, Miroslav, and Stan Matwin. "Addressing the curse of imbalanced training sets: one-sided selection." Icml. Vol. 97. (1997)
13. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F. Partitioning and Local Matching Learning of Large Biomedical Ontologies : ACMSIGAPP SAC, Limassol, Cyprus, april 2019 (to appear)
14. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F. (2018). OAEI 2018 results of POMap+. Ontology Matching, 192.
15. Porter, Martin F. "Snowball: A language for stemming algorithms." (2001)
16. Lucy Lu Wang et al. Ontology Alignment in the Biomedical Domain Using Entity Definitions and Context. (2018)
17. Müllner, Daniel. "Modern hierarchical, agglomerative clustering algorithms." arXiv preprint arXiv:1109.2378 (2011)
18. More, Ajinkya. "Survey of resampling techniques for improving classification performance in unbalanced datasets." arXiv preprint arXiv:1608.06048 (2016)
19. Nezhadi, Azadeh Haratian, Bita Shadgar, and Alireza Osareh. "Ontology alignment using machine learning techniques." IJCSIT (2011)
20. Ngo, DuyHoa, and Zohra Bellahsene. "Overview of YAM++—(not) Yet Another Matcher for ontology alignment task." Web Semantics: Science, Services and Agents on the World Wide Web 41 (2016)
21. NKISI-ORJI, I., WIRATUNGA, N., MASSIE, S., HUI, K.-Y. and HEAVEN, R. Ontology alignment based on word embedding and random forest classification. ECML PKDD 2018, Dublin, Ireland. (2018)
22. Shvaiko, Pavel, Jérôme Euzenat, Ernesto Jiménez, Michelle Cheatham, and Oktie Hassanzadeh. OM 2017, International Workshop on Ontology Matching, (2017)
23. Stuckenschmidt, Heiner, Christine Parent, and Stefano Spaccapietra, eds. Modular ontologies: concepts, theories and techniques for knowledge modularization. Vol. 5445. Springer, 2009.
24. Tomek, Ivan. "Two modifications of CNN." IEEE Trans. Systems, Man and Cybernetics 6 (1976)
25. Wilson, Dennis L. "Asymptotic properties of nearest neighbor rules using edited data." IEEE Transactions on Systems, Man, and Cybernetics 3 (1972)
26. Xue, Xingsi, and Jeng-Shyang Pan. "A segment-based approach for large-scale ontology matching." Knowledge and Information Systems 52.2 (2017)