



Building an effective and efficient background knowledge resource to enhance ontology matching

Amina Annane, Zohra Bellahsene, Faical Azouaou, Clement Jonquet

► To cite this version:

Amina Annane, Zohra Bellahsene, Faical Azouaou, Clement Jonquet. Building an effective and efficient background knowledge resource to enhance ontology matching. *Journal of Web Semantics*, 2018, 51, pp.51-68. 10.1016/j.websem.2018.04.001 . hal-01809627v2

HAL Id: hal-01809627

<https://hal.science/hal-01809627v2>

Submitted on 4 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Building an effective and efficient background knowledge resource to enhance ontology matching

Amina Annane^{1,2}, Zohra Bellahsene², Faïçal Azouaou¹, Clement Jonquet^{2,3}

¹ *Ecole nationale Supérieure d'Informatique, BP 68M, 16309, Oued-Smar, Alger, Algérie*

² *LIRMM, Université de Montpellier, CNRS, Montpellier, France*

³ *Center for BioMedical Informatics Research (BMIR), Stanford University, USA*

Abstract

Ontology matching is critical for data integration and interoperability. Original ontology matching approaches relied solely on the content of the ontologies to align. However, these approaches are less effective when equivalent concepts have dissimilar labels and are structured with different modeling views. To overcome this semantic heterogeneity, the community has turned to the use of external background knowledge resources. Several methods have been proposed to select ontologies, other than the ones to align, as background knowledge to enhance a given ontology-matching task. However, these methods return a set of complete ontologies, while, in most cases, only fragments of the returned ontologies are effective for discovering new mappings. In this article, we propose an approach to select and build a background knowledge resource with just the right concepts chosen from a set of ontologies, which improves efficiency without loss of effectiveness. The use of background knowledge in ontology matching is a double-edged sword: while it may increase recall (i.e., retrieve more correct mappings), it may lower precision (i.e., produce more incorrect mappings). Therefore, we propose two methods to select the most relevant mappings from the candidate ones: (1) a selection based on a set of rules and (2) a selection based on supervised machine learning. Our experiments, conducted on two Ontology Alignment Evaluation Initiative (OAEI) datasets, confirm the effectiveness and efficiency of our approach. Moreover, the F-measure values obtained with our approach are very competitive to those of the state-of-the-art matchers exploiting background knowledge resources.

Keywords: Ontology matching, Ontology alignment, Background knowledge, Indirect matching, External resource, Anchoring, Derivation, Background knowledge selection, Supervised machine learning.

1. Introduction

Ontologies provide conceptual models to represent and share knowledge. Some of the data management challenges for which ontologies are often used include interoperability [55] and data integration [35]. In recent years, because of the large number of ontologies developed, especially in domains that produce and manage an increasing amount of data (such as biomedicine [43]), these challenges have become increasingly complex. To achieve interoperability and integration, one solution is to identify mappings (correspondences) between different ontologies of the same domain. This process is known as *ontology matching* or *ontology alignment*.

Ontologies are heterogeneous because they have been designed independently, by different developers, and following diverse modeling principles and patterns. This heterogeneity makes the matching process complex [20]. The

first ontology matching methods were based only on the lexical and structural content of the ontologies to align; this is known as *direct matching* or *content-based matching*. To that end, many syntactic and structural similarity measures have been developed [8, 42, 20]. However, direct matching is less effective to find correspondences between concepts that are equivalent, but described with dissimilar labels and structured with different modeling views [1, 50].

To overcome this semantic heterogeneity, the community has turned to the exploitation of external knowledge resource(s), commonly called background knowledge resources. In contrast to direct matching, this approach is known as *indirect matching*, *BK-based matching* or *context-based matching* [37], as it exploits external resources to identify mappings between the ontologies to align.

The BK-based matching approach raises two main issues: (i) how to select (or build) background knowledge resource(s) for a given ontology matching task? and (ii) how to concretely use the selected background knowledge resource(s) to enhance the quality of the matching result?

In the literature, several works have addressed these

Email addresses: a_annane@esi.dz (Amina Annane^{1,2}), bella@lirmm.fr (Zohra Bellahsene²), f_azouaou@esi.dz (Faïçal Azouaou¹), jonquet@lirmm.fr (Clement Jonquet^{2,3})

two issues jointly or separately. The selection of background knowledge resources consists in choosing m resources among the n possible ones [23, 27, 9, 37]. Although only fragments of the m resources may actually prove effective for discovering new mappings, entire resources are selected. In addition, almost all previous works exploit the selected resources independently of each other [46, 25], whereas, as we will show in this paper, more correct mappings may be identified when we combine the selected resources into a single one.

In this article, we propose a novel approach to build a customized background knowledge resource from a set of ontologies to derive equivalence mappings. This approach is inspired from our previous work [3]. The built resource reduces the computation-time cost of the BK-based matching approach (i.e., efficient), and allows to improve the quality of the alignments generated by the direct matching (i.e., effective).

In [3], we considered the mappings stored in the NCBO BioPortal repository as background knowledge [43]. However, we cannot find such mapping repository in every domain. Therefore, in this article, we tried to make our approach more generic taking a set of ontologies as input and using an automatic matcher to extract the mappings to be used as background knowledge.

The use of external knowledge resources in ontology matching is a double-edged sword. Indeed, though these resources provide new information to find correct mappings, incorrect mappings may also be generated [37]. Consequently, selecting correct mappings from the candidate ones is particularly challenging in the context of BK-based matching. In this article, we propose two new selection methods. The first one is based on a set of rules, while the second one is based on the supervised machine learning technique. To enable the use of a classification machine learning algorithm, we designed a set of 27 attributes based on the built background knowledge resource.

We performed extensive experiments on two datasets taken from OAEI¹, with two sets of preselected ontologies, to evaluate the performance of our approach. The experiment results confirm the efficiency and effectiveness of our approach. Moreover, we compared our results to those of the state-of-the-art systems that exploit background knowledge resources. Our F-measure values are very competitive relative to the best ones reported in the literature.

To sum up, the main contributions of this article are:

- A formalized general workflow for BK-based ontology matching;
- A novel and dynamic approach to building a background knowledge resource from a set of ontologies;
- A BK-based matching approach that uses the built background knowledge resource;

- Two final mapping selection methods: a rule-based method, and a machine-learning based one.
- Extensive experiments on two OAEI datasets to demonstrate the efficiency of the built background knowledge resource, and the effectiveness of our approach.

The remainder of this paper is organized as follows. In Section 2, we define the basic notions used herein and provide a brief overview of our approach. We then describe the three main steps of our approach in detail. In Section 3, we review the selection and building of the background knowledge resource. In Section 4, we show how the built resource is used to derive candidate mappings. In Section 5, we describe our methods for selecting final mappings. In Section 6, we discuss the efficiency of our approach. In Section 7, we present the experimental material used for the evaluation. In Section 8, we present our results for OAEI Anatomy and LargeBio datasets. In Section 9, we discuss some limitations of our work. In Section 10, we provide a summary of related work. Finally, in Section 11, we conclude and discuss some future research.

2. Preliminaries

In this section, we define the various concepts underlying our approach.

2.1. Ontology matching

Ontology matching can be formally defined as a function that takes two ontologies O_s and O_t , a set of parameters P , and a set of resources R , and returns an alignment A between O_s and O_t .

An **Alignment** of ontologies O_s and O_t is a set of mappings between their different entities e_s and e_t .

A **Mapping** (or a correspondence) between an entity e_s (e.g., property, class) belonging to ontology O_s and an entity e_t belonging to ontology O_t is a four-tuple of the form: $m = \langle e_s, e_t, r, k \rangle$ where:

- r is a relation such as equivalence (\equiv), more general (\supseteq), less general (\sqsubseteq), etc.
- k is a confidence score (typically in the $[0, 1]$ range) holding for the correspondence between the entities e_s and e_t .

A **Similarity measure** is a function $f : E_s \times E_t \rightarrow [0..1]$ where E_s is the set of O_s entities and E_t is the set of O_t entities. For each pair of entities (e_s, e_t) , a similarity measure computes a real number, generally between 0 and 1, expressing the similarity between the two entities. There are several kinds of similarity measures: syntactic, semantic and structural [8, 44, 42].

The previous definitions were adopted from [19].

¹<http://oaei.ontologymatching.org/>

A **Matcher** is a matching algorithm that implements one similarity measure or combines several to discover mappings between ontologies. The term *Matcher* is also used in the literature to identify an ontology matching system [15].

2.2. Ontology matching evaluation

Evaluating a given alignment is usually done with three measures: precision, recall and F-measure [11]. These measures are computed with respect to a reference alignment that contains all the correct mappings. Precision is defined as the number of correctly identified mappings divided by the total number of mappings found (correct + incorrect). Recall is defined as the number of correctly identified mappings divided by the number of all possible correct mappings (the size of the reference alignment). A perfect precision score of 1.0 means that every mapping returned by the matcher is correct; Precision measures correctness. A perfect recall score of 1.0 means that all correct mappings were returned; Recall measures completeness. The F-measure is the harmonic mean of precision and recall. It measures the overall accuracy of an alignment. Let A be an alignment produced by a given matcher and R the reference alignment. Precision, Recall and F-measure are computed as follows:

$$\begin{aligned} \text{Precision} &= \frac{|A \cap R|}{|A|} \\ \text{Recall} &= \frac{|A \cap R|}{|R|} \\ \text{F-measure} &= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

2.3. Background knowledge

In the context of ontology matching, there is no commonly accepted or strict definition of what background knowledge is. We define it as any set of external knowledge resources that provides lexical or semantic information about the domain(s) of the ontologies to align or some of the entities therein. It could be any datasets related to the ontologies to align, other ontologies than the ones to align, other previously generated mappings, lexical resources, the Web, etc.

In this article, we use the acronym **BK** to refer to a background knowledge resource used within the matching process. For instance, if such a resource is an ontology, we will call it a *BK ontology*. Similarly, the expression *BK-based method* denotes a method that exploits a background knowledge resource.

2.4. Supervised machine learning

Supervised machine learning is the task of automatically inferring a function from training data [40]. The learned function $f : x \rightarrow y$ maps the input object x to an output y . When using the machine learning technique for a classification task, the learned function is called a *classifier*. The input x is composed of a set of attribute

values that describe the object to classify, while the output y is the class in which the object x will be classified by the learned classifier. The training data is a set of objects already classified (containing both attributes and class), while the test data is the set of objects to classify. A supervised machine learning algorithm analyzes the training data and produces a classifier that will be used to classify the test data objects.

2.5. Overview of our approach

As shown in Figure 1, the general workflow of our BK-based matching approach includes three main steps. The first step consists in selecting a BK from an initial set of ontologies. In our approach, we do not select complete ontologies; instead we select concepts from the initial ontologies. We then combine the selected concepts in one single resource that we call the **built BK**. In the second step, we exploit the built BK to generate all possible candidate mappings between the ontologies to align. The last step consists in selecting the most relevant candidate mappings to produce the final alignment.



Figure 1: Main steps of our BK-based ontology matching approach.

3. BK Selection

We can formally define BK selection as a function that takes a set of knowledge resources KR , a set of ontologies to align O , and optionally, a set of parameters P (e.g., threshold values), and returns the BK that will be used in the matching process. The returned BK may be a subset of KR [23], or a novel resource built from KR (our approach).

For ontology matching, the automatic selection of ontologies as background knowledge has been proposed in several works [50, 46, 27, 23]. However, all these methods return complete ontologies as a final BK for the matching process. Our hypothesis is that within each BK ontology, especially large ones, only fragments may actually prove effective. Hence, the issue is that of the selection of these fragments from each BK ontology and their combination to build an effective and efficient BK. In our approach, we tackle this issue by selecting only the relevant concepts from the initial ontologies related to the matching task. We then combine these selected concepts to build the BK. As we will experimentally demonstrate, the built BK has a reduced size comparing to the preselected ontologies size, which improves the efficiency of the BK-based matching without loss of effectiveness. Furthermore, the built BK interconnects concepts from different preselected ontologies via mappings, thereby generating mappings across several intermediate ontologies.

In the following, we detail the four steps involved in the BK selection process (see Figure 2).

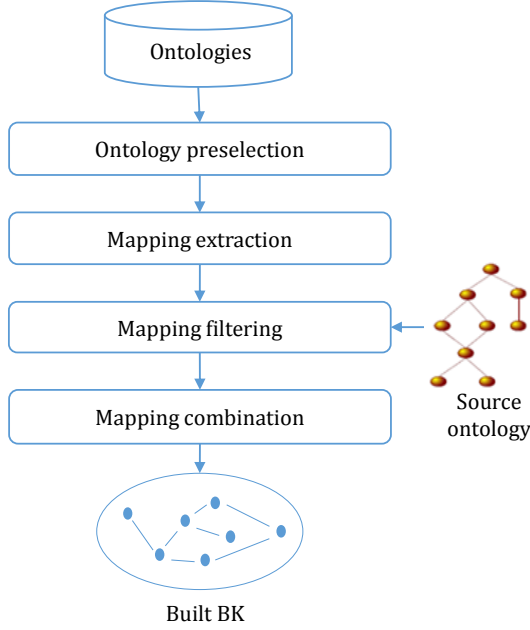


Figure 2: Overview of the BK selection process.

3.1. Ontology preselection

Today, a simple Google Search for "filetype:owl" returns around 34K results. Fittingly, these ontologies are often organized per domain or community in ontology libraries [56, 10] such as the NCBO BioPortal, the AgroPortal [33] or the Marine Metadata Interoperability repository [49]. Ontology preselection consists in determining which ontologies to consider for the BK selection process among all the ontologies that exist. It aims at reducing the search space for the BK selection process by eliminating at the outset ontologies that would not be effective to identify new mappings (e.g., ontologies that are not of the same domain as the ontologies to align). The preselected ontologies may be an ontology repository [9], a specific set of ontologies [23, 22, 27] or all ontologies indexed by a given semantic web search engine [50, 37].

In the related works, ontology preselection has not been formalized as a step of the BK-based ontology matching workflow, except in [37] where ontology preselection was called *ontology arrangement*.

In this article, we do not focus on automating ontology preselection. In our approach, and, at the best of our knowledge, in all related works, ontology preselection is performed manually [50, 23, 22, 27, 37].

3.2. Mapping extraction

The experiments reported in [29, 50, 37] showed that combining several BK ontologies generates more correct mappings. Figure 3 illustrates this benefit, sampled from our evaluation. Each concept is represented with the term: ontology#ConceptIdentifier and interconnected with mappings. The source and target concepts are linked via at least two intermediate concepts which belong to two different BK ontologies. Such correct mapping would not have

been identified if we had used each intermediate ontology separately from the others (one intermediate concept at a time). Therefore, in this step, we extract all possible mappings between the preselected ontologies to be able to generate mappings across several intermediate ontologies.

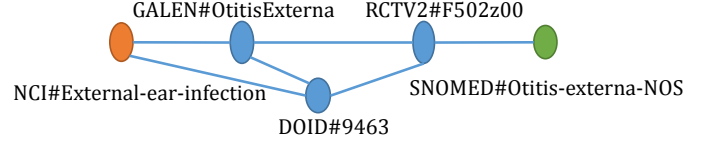


Figure 3: Example of a correct mapping between NCI and SNOMED derived across intermediate concepts from different BK ontologies.

Let $S = \{O_1, O_2, \dots, O_n\}$ be the set of preselected ontologies. In this step, each ontology O_i in S is matched to the other preselected ontologies that have a higher index (i.e., $O_{i+1}, O_{i+2}, \dots, O_n$). The matching of each couple of ontologies (O_i, O_j) provides an alignment that is a set of s mappings $A_{ij} = \{m_1, m_2, \dots, m_s\}$. For n preselected ontologies, the result is the union of $\sum_{i=1}^{n-1} (n-i)$ alignments. More specifically, the result is the union of all mappings that compose the different alignments A_{ij} : $M = \bigcup_{i=1}^{n-1} \bigcup_{j=i+1}^n A_{ij}$.

The easiest way to extract these mappings is to use an automatic matcher. Several state-of-the-art matchers, such as YAM++ [41], LogMap [30], AML [24], etc., are readily available. As shown in previous OAEI campaigns, these systems provide high-quality alignments (i.e., alignments with high F-measure score). Furthermore, if available, mappings between the preselected ontologies that are manually created or human-curated should be added to the automatically extracted ones. For instance, in the biomedical domain, cross-references between OBO Foundry ontologies [52] may be considered as manual mappings. Note that the mapping extraction task may be ignored if the preselected ontologies are not to be combined.

3.3. Mapping filtering

The preselected ontology concepts likely to generate new mappings should be related directly or indirectly to the source ontology. Conversely, those concepts not related to the source ontology will not help generate new mappings. Hence, it seems more efficient to eliminate the latter at the outset.

We start by matching the source ontology O_s to the preselected ontologies in S . In order to improve efficiency, the smallest of the ontologies to align is chosen as the source ontology. The mappings obtained by matching the source ontology to the preselected ontologies initialize the set of filtered mappings, noted FM . Recursively, we enrich FM by selecting all the mappings in M related to the target concepts of mappings already present in FM , and so on, until no new mapping is found in M . More precisely, until all mappings related to the source ontology in M are in FM . In each step, FM is enriched as follows:

$FM = FM \cup \{m_i/m_i \in M \text{ and } C_s(m_i) = C_t(m_j) \text{ and } m_j \in FM\}$

where $C_s(m_i)$ is a function that returns the source concept of the mapping m_i and $C_t(m_j)$ is a function that returns the target concept of the mapping m_j .

3.4. Mapping combination

Mappings filtered in the previous step are then combined in one unique graph where nodes are concepts and edges are mappings that link these concepts. This combination insures that each concept appears only once (i.e., mappings that share a concept are merged). Figure 4 shows an example of mapping combination. m_1 and m_2 are two mappings that have a common concept e_2 . The combination keeps only one occurrence of the concept e_2 . Note that, thanks to this combination, concepts that are not directly connected (e_1 and e_3 in Figure 4) may be indirectly connected through common concepts.

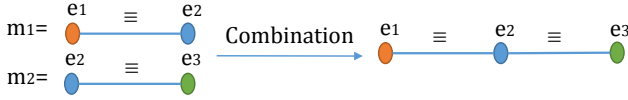


Figure 4: Mapping combination example.

In the resulting graph, each selected concept (node) is described with four attributes: (i) the URI of the concept, (ii) the URI of the ontology to which the concept belongs, (iii) the preferred label of the concept and (iv) the concept synonyms. The mappings (or edges) between concepts are described with three attributes: (i) the source from which each mapping has been extracted. It may be the name of a resource such as UMLS, or the ontology matching tool name when the mapping was generated automatically. (ii) the mapping score and (iii) the type attribute that indicates whether the mapping was generated manually or automatically. This graph is the *built BK* that will be exploited in the BK use step.

4. BK Use

In this step, we use the built BK to derive candidate mappings between the ontologies to align. Figure 5 illustrates this part of the process, which includes (i) anchoring and (ii) deriving candidate mappings.

4.1. Anchoring

Anchoring consists in localizing the entities of the ontologies to align in the background knowledge resource [2, 50]. In our case, this is done by a direct matching between the ontologies to align and the built BK. Anchoring mappings are then added to the built BK graph.

Let M be a matcher, O_s and O_t two ontologies to align, BBK the built BK, e_s an entity belonging to O_s , e_t an entity belonging to O_t and e'_s, e'_t entities belonging to BBK . Anchoring consists in producing two alignments A_s and A_t with the matcher M , where:

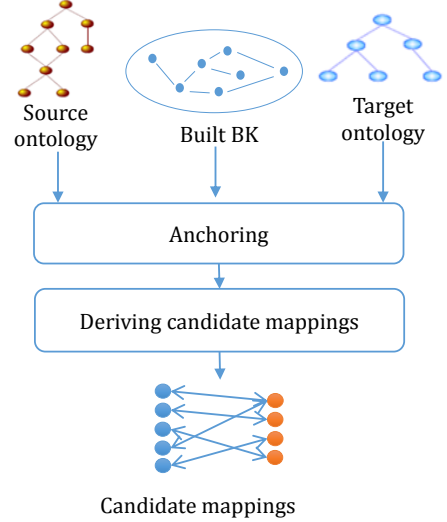


Figure 5: Overview of BK use process.

- $A_s = M(O_s, BBK)$ a set of mappings of the form $m = \langle e_s, e'_t, r, k \rangle$
- $A_t = M(BBK, O_t)$ a set of mappings of the form $m = \langle e'_s, e_t, r, k \rangle$

BBK entities e'_s and e'_t that appear in A_s and A_t are called *anchors*, while e_s and e_t are called *anchored entities*.

Note that, for the source ontology, we may simply reuse the mappings produced in the mapping filtering step between the source ontology and the preselected ontologies. This is feasible when both steps use the same matcher. However, BK selection and BK use can be two completely independent steps.

4.2. Deriving candidate mappings

In this step, candidate mappings are derived between the ontologies to align using the graph structure of the built BK. We search for each source concept anchored to the built BK, all paths leading to the target ontology concepts. Each path found may be represented by a set of n mappings as follows:

$$P = \{\langle e_{s1}, e'_{t1}, r_1, k_1 \rangle, \langle e'_{s2}, e'_{t2}, r_2, k_2 \rangle, \dots, \langle e'_{sn}, e_{tn}, r_n, k_n \rangle\}$$

Where e_{s1} belongs to the source ontology, e_{tn} belongs to the target ontology and $e'_{ti} = e'_{s(i+1)}$. Each path found provides a candidate mapping $\langle e_{s1}, e_{tn}, r, k \rangle$. r results from the composition of the different r_i on the path P . Similarly, k results from the composition of the different k_i on the path P . In this article, we only deal with equivalence mappings (i.e., all r_i are equivalences). However, our approach may be extended to other kinds of mapping relationships, provided a strategy to compose different relationships on the same path is defined [18]. Note that the intermediate concepts of a given path originate from different ontologies, which represents a derivation across several intermediate ontologies.

In ontology matching, the objective of using background knowledge resources is to complement direct matching but

not to replace it. Indeed, direct matching may identify mappings that can be missed in BK-based matching and vice versa. Therefore, to complement the set of candidate mappings, we propose adding direct-matching mappings to the set of derived candidate mappings.

5. Final mapping selection

To select the most accurate mappings, an effective mapping selection method must be used. Candidate mappings consist in a set of paths linking the source to the target concepts. Several paths may represent the same candidate mapping. Thus, to compute the final score k for a given candidate mapping, we must address two issues:

1. How to **compose** the different mapping scores of the same path?
2. How to **aggregate** the scores of different paths representing the same candidate mapping?

Related work suggested to use algebraic functions, such as multiplication, average, maximum, etc. to compose different mapping scores [39]. These functions may also be used for aggregation (issue 2).

In the following, we use the term *configuration* for a given pair of composition and aggregation functions. For instance, the multiplication-maximum configuration means that the composition (issue 1) is the multiplication of the path scores, while the aggregation (issue 2) is performed with the maximum function. For a given candidate mapping, we may compute one or multiple scores according to the selection method. Indeed, different configurations return different scores for the same candidate mapping.

5.1. Rule-based selection

Rule-based selection of the final mappings consists in defining a set of rules to decide whether or not to keep a given candidate mapping in the final alignment. In our method, we propose the following rules:

1. Mappings returned by direct and indirect matching are selected.
2. Mappings resulting from the composition of only manual mappings are selected.
3. For each source concept, the target candidate with the highest mapping score is retained.
4. For each target concept, the source candidate with the highest mapping score is retained.

For rules 3 and 4, the score may be controlled by a given threshold. The score of the candidate mappings is computed with the multiplication-maximum configuration.

5.2. Machine learning-based selection

As previously discussed, there exist multiple possible algebraic function configurations to compose mapping scores of the same path, and to aggregate scores of different paths representing the same candidate mapping. However, testing the performance of all possible configurations to find the most suitable one for a given matching task is fastidious. Additionally, finding the best configuration for a given matching task does not amount to finding it for all matching tasks. Furthermore, one may combine several configurations to improve the effectiveness of the selection method; for example, one could combine average-multiplication, maximum-multiplication and average-average configurations. Indeed, each configuration may provide a piece of information which could help to select the most relevant mappings. In this case, however, we would also have to define how to combine the different values of these configurations to select the final mappings. This renders the task even more complex.

Supervised Machine Learning technique (ML) is an appropriate option to address this issue. Indeed, according to the training data, ML automatically customizes a classification function (classifier) that combines several attributes (selection variables). We therefore propose to cast the problem of mapping selection into a classification problem as follows:

- The test data are the candidate mappings between the source and target ontologies to be classified as *true* or *false*.
- The training data are a set of candidate mappings already classified as *true* or *false*. These candidate mappings are completely distinct from the test data (the candidate mappings to classify).
- The attributes that describe each candidate mapping are the different configurations and any variable that can help to classify a given candidate mapping.

In the following, we present the candidate mapping attributes, the training data as well as *RandomForest*, the machine learning algorithm used in this article.

5.2.1. Candidate mapping attributes

In our case, the attributes are the selection variables. Indeed, each attribute is a decision variable that will help to decide if a given candidate mapping will be classified as *true* or *false*. In related work, to classify the candidate mappings, similarity measures between source and target concepts were used. Here, however, the candidate mappings are a set of paths between source and target concepts. Therefore, we need to define new attributes. We thus propose a set of 27 selection attributes for each candidate mapping:

Direct score: if the candidate mapping belongs to the alignment returned by the direct matching, the direct

score is the score of the candidate mapping in this alignment; otherwise, it is 0. Our intuition is that the mappings returned by the direct matching are likely to be correct.

Number of paths representing the candidate mapping: in fact, candidate mappings returned by many paths are more likely to be correct than those returned by few paths.

Path length attributes: for each candidate mapping, we compute three attributes that are (i) the minimum length, (ii) the maximum length and (iii) the average length of paths that represent the candidate mapping. Our intuition is that, the shorter the paths, the more relevant the candidate mapping will be.

Mapping score attributes: For each candidate mapping, 21 score attributes are computed. Indeed, for each path that represents the candidate mapping, we compute seven values with the following composition functions: (1) maximum, (2) minimum, (3) average, (4) multiplication, (5) sum, (6) variance and (7) average divided by variance. Each function takes the scores of the mappings that make this path as an input. We then aggregate path scores for each composition function with three functions: (1) maximum, (2) minimum and (3) average. For instance, when using variance as a composition function, we compute three attributes from the paths that represent the candidate mapping with the following configurations: maximum-variance, minimum-variance and average-variance. We repeat this process with the other six composition functions to obtain 21 attributes.

Maximum average of manual mappings: For each path representing the candidate mapping, we compute the average number of manual mappings (i.e., the number of manual mappings divided by the number of mappings of this path). Then, the maximum average is taken as an attribute. Indeed, paths containing manual mappings are more relevant than those containing only automatic mappings.

Let us take an example to illustrate the computation of the various attributes. Figure 6 shows an actual example from our evaluation (described further). Concepts are represented in the form: ontology#ConceptIdentifier; the values on edges are the mapping scores returned by the automatic matcher; OBO is a manual mapping. As we can see, the source concept is anchored to three BK concepts, while the target concept is anchored to only one. The derivation step returns four paths linking the source concept to the target concept.

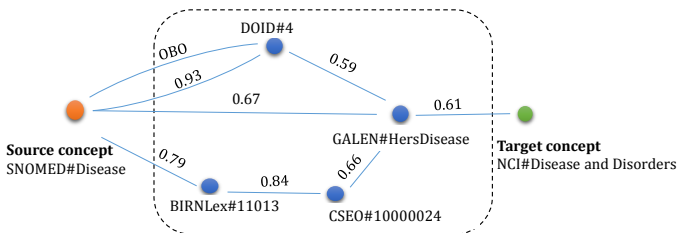


Figure 6: Example of candidate mapping derivation.

The following candidate mapping: (SNOMED#Disease, NCI#Disease and Disorders) is described by the following attributes:

Direct mapping score: 0; number of paths: 4; Average path length: $(3 + 3 + 2 + 4)/4 = 3$; minimum path length: 2; maximum path length: 4; Average manual mapping: $1/3$ because there is only one path of length 3 that contains one manual mapping; For score attributes, we illustrate one (multiplication) of the seven composition functions proposed. We start by computing a score for each path, as shown in Table 1. Then, using these path scores, we compute the following attributes: maximum scores: 0.41; minimum scores: 0.27; average scores: $(0.36+0.33+0.41+0.27)/4=0.34$.

Table 1: Path scores for Figure 6 example.

Path nodes	Score
DOID#4, GALEN#HersDisease	$1*0.59*0.61=0.36$
DOID#4, GALEN#HersDisease	$0.93*0.59*0.61=0.33$
GALEN#HersDisease	$0.67*0.61=0.41$
BIRNLex#11013, CSEO#10000024, GALEN#HersDisease	$0.79*0.84*0.66*0.61=0.27$

5.2.2. Training data

In our case, training data are candidate mappings annotated by *true* (correct mapping) or *false* (incorrect mapping) and described by all the previously presented attributes. As is usual with supervised machine learning, obtaining training data requires previously generated and curated reference alignments from other ontologies than those to align. Preferably, the aligned ontologies are of the same domain as the ontologies to align. To obtain the training data, we propose to apply our approach to the aligned ontologies (i.e., BK selection and BK use). We then, compute the 27 attributes for each derived candidate mapping and annotate it by *true* or *false* according to the reference alignments of the aligned ontologies (see Figure 7).

5.2.3. RandomForest machine learning algorithm

There are several algorithms for learning a classification function from a set of training data. In our experiments, we used *RandomForest*, a non-linear method for classification [6]. In the training step, it learns a multitude of decision trees by creating a different random subset to train each decision tree. In the classification step, it aggregates the results of these trees by outputting the most frequent class. Due to this strategy, Random Forrest has the advantage of being efficient on any type of data set. Our choice of this algorithm was motivated by its performance in preliminary experiments. Indeed, we evaluated the classification results produced by different ML

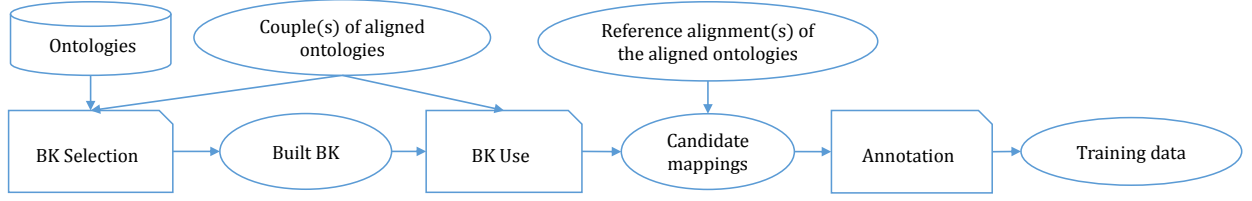


Figure 7: Training data generation process.

algorithms implemented in the Weka framework [26] such as trees algorithm (J48, RandomForest, RandomTree) and rules algorithms (JRIP, oneR, etc.); RandomForest generated the best results. This confirms the results reported in [28]: for learning linkage rules, the non-linear classifiers (trees) are the most appropriate.

6. Efficiency gain with the built BK

Building a new resource (i.e., the built BK) from the preselected ontologies is more efficient than returning complete ontologies as background knowledge. In this section, we estimate the computation time of our BK selection approach and that of the traditional approach, then we compare them to demonstrate the efficiency of our approach.

The traditional approach refers to the BK selection methods that match the source and target ontologies to all the preselected ontologies, and then they use the generated alignments to select the ontologies to be exploited as background knowledge [27, 23, 37].

Anchoring is the step that follows BK selection, its computation time depends on the selected BK: a set of ontologies or the built BK. Therefore, we include the anchoring computation time in our comparison. However, we do not include the mapping extraction computation time because it is performed once between the preselected ontologies independently of the matching tasks. Moreover, when comparing our approach to those that use each BK ontology separately (derivation across only one intermediate concept) [27, 23], the mapping extraction time has a zero value. Indeed, these works do not match BK ontologies between each other.

Let $KR = \{O_1, O_2, \dots, O_n\}$ be the set of preselected ontologies, O_S the source ontology and O_T the target ontology, $t(M, O_1, O_2)$ the function that returns the time required by the matcher M to align the ontologies O_1 and O_2 . When using the traditional approach, the selected BK is a set of k ontologies $SR = \{SO_1, \dots, SO_k\}$, with $SR \subseteq KR$. However, when using our approach, the selected BK is one resource built from KR ontologies, called BBK . The BK selection computation-time is computed as follows.

- Traditional approach:

$$T_1 = \sum_{i=1}^n t(M, O_S, O_i) + \sum_{i=1}^n t(M, O_T, O_i) + \alpha.$$

- Our approach:

$$T'_1 = \sum_{i=1}^n t(M, O_S, O_i) + \beta.$$

Where α and β are the computation time required for the treatments performed after the BK selection matching tasks. In the traditional approach, it may be the time of computing similarity measures and ranking the preselected ontologies [27, 23, 37]. In our approach, it is the time of selecting the mappings related to the source ontology and combining them. Usually, the values of α and β are negligible comparing to that of the matching tasks performed within the BK selection process.

In the example illustrated in Figure 8, with four preselected ontologies, the traditional approach performs eight matching tasks generating the alignments A_1 to A_8 , while our approach performs four matching tasks generating the alignments A_1 to A_4 .

For the anchoring step, we distinguish two cases:

Case 1: Reusing BK selection alignments as anchoring alignments.

The anchoring computation time is computed as follows.

- Traditional approach: the anchoring alignments are already available, no additional matching task is necessary.

$$T_2 = 0.$$

- Our approach: The BK selection alignments are related only to the source ontology. Hence, another matching task is necessary to anchor the target ontology to the BBK (e.g., the time necessary to generate the alignment B_2 in Figure 8 (b)).

$$T'_2 = t(M, O_T, BBK).$$

The computation time of BK selection and anchoring is estimated as follows.

- Traditional approach:

$$T = T_1 + T_2 =$$

$$\sum_{i=1}^n t(M, O_S, O_i) + \sum_{i=1}^n t(M, O_T, O_i) + \alpha. \quad (1)$$

- Our approach:

$$T' = T'_1 + T'_2 =$$

$$\sum_{i=1}^n t(M, O_S, O_i) + \beta + t(M, O_T, BBK). \quad (2)$$

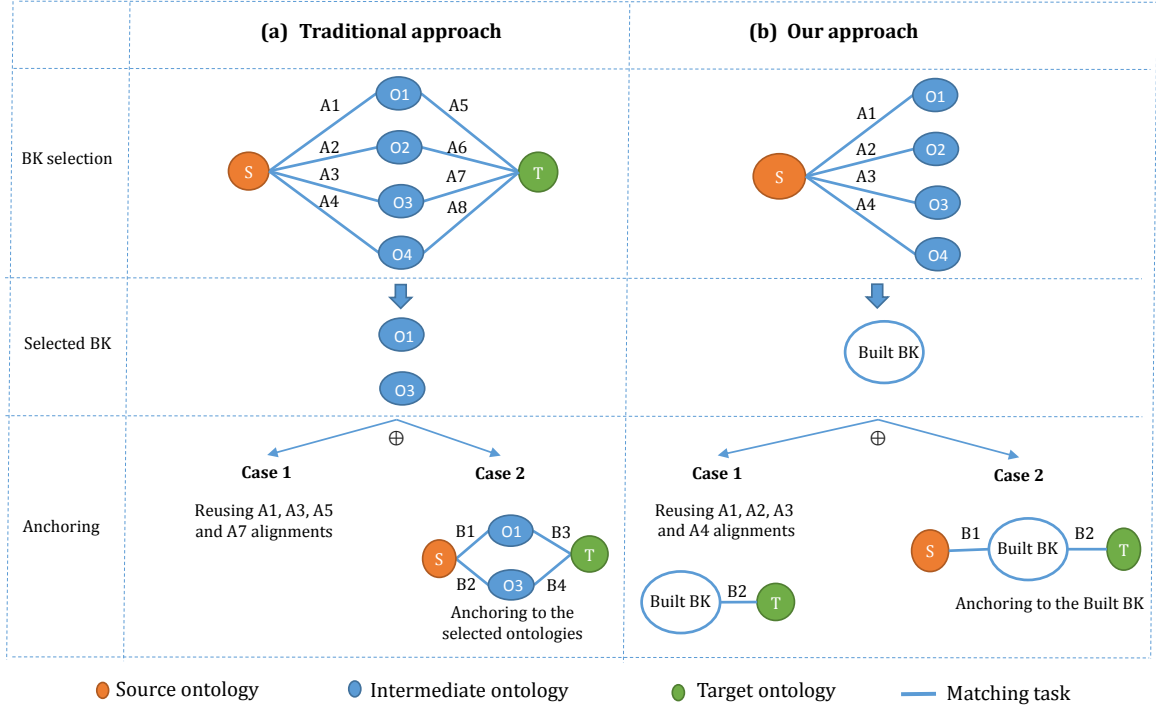


Figure 8: BK Selection and anchoring: Traditional approach vs. our approach

- Comparison: Traditional approach vs. our approach

$$T - T' =$$

$$\sum_{i=1}^n t(M, O_T, O_i) - t(M, O_T, BBK) + (\alpha - \beta). \quad (3)$$

Intuitively the difference $T - T'$ is always positive. Indeed, the difference $(\alpha - \beta)$ tends to zero, and matching the target ontology to all the preselected ontologies takes more much time than matching the target ontology to the BBK. This intuition is validated with experiments in Section 8.4.2.

Case 2: The source and target ontologies are anchored to the selected BK with another matcher M' .

In our approach, the anchoring step requires two matching tasks, while in the traditional approach, the number of matching tasks depends on the number of the selected BK ontologies. For instance, in Figure 8 (a), with two selected BK ontologies, four matching tasks are necessary to generate B_1 to B_4 . Thus, the anchoring computation time is computed as follows.

- Traditional approach:

$$T_2 = \sum_{j=1}^k t(M', O_S, SO_j) + \sum_{j=1}^k t(M', O_T, SO_j).$$

- Our approach:

$$T'_2 = t(M', O_S, BBK) + t(M', O_T, BBK).$$

The computation time of BK selection and anchoring is estimated as follows.

- Traditional approach:

$$T = T_1 + T_2 =$$

$$(1) + \sum_{j=1}^k t(M', O_S, SO_j) + \sum_{j=1}^k t(M', O_T, SO_j).$$

- Our approach:

$$T' = T'_1 + T'_2 = (2) + t(M', O_S, BBK).$$

- Comparison: Traditional approach vs. our approach

$$T - T' = (3) + \sum_{j=1}^k t(M', O_S, SO_j) +$$

$$\sum_{j=1}^k t(M', O_T, SO_j) - t(M', O_S, BBK).$$

Our hypothesis is that the difference $T - T'$ is always positive. Indeed, the formula (3) is positive as explained in **Case 1**, and matching the ontologies to align to the selected BK ontologies (i.e., $\sum_{j=1}^k t(M', O_S, SO_j) + \sum_{j=1}^k t(M', O_T, SO_j)$) takes more time than matching the source ontology to the BBK (i.e., $t(M', O_S, BBK)$). Note that, in our approach, matching the target ontology to the BBK (e.g., generating B_2 in Figure 8 (b)) is common to the two cases, and its computation time is already included in the formula (3). We discussed this case at the end of Section 8.4.2.

7. Experiment materials

7.1. Evaluation datasets

To evaluate our approach, we chose two OAEI tracks: Anatomy and Large biomedical ontology (LargeBio). Our choice was motivated by the fact that, for these tracks,

Table 2: LargeBio track (the last column shows the number of mappings in the reference alignment provided by OAEI).

Task #	Task name	#Source concepts	#Target concepts	#Mappings
Task 1	FMA-NCI small fragments	3,696	6,488	2,686
Task 2	NCI-FMA Whole ontologies	66,724	78,989	2,686
Task 3	FMA-SNOMED small fragments	10,157	13,412	6,026
Task 4	FMA whole with SNOMED large fragment	78,989	122,464	6,026
Task 5	NCI-SNOMED small fragments	23,958	51,128	17,210
Task 6	NCI whole with SNOMED large fragment	66,724	122,464	17,210

the state-of-the-art systems have used ontologies as background knowledge to enhance the quality of their alignments. Hence, evaluating using these tracks with the same preselected ontologies allowed us to compare our results to the state-of-the-art’s results.

7.1.1. Anatomy OAEI track

The Anatomy track consists in finding an alignment of 1,516 mappings between the Adult Mouse Anatomy (2,744 classes) and a subset of the National Cancer Institute (NCI) Thesaurus (3,304 classes) describing human anatomy [14].

7.1.2. LargeBio OAEI track

The Large Biomedical (LargeBio) OAEI track² aims at finding alignments between several large and semantically rich biomedical ontologies: the Foundational Model of Anatomy (FMA) [48], National Cancer Institute Thesaurus (NCI) [51] and SNOMED Clinical Terms (SNOMED-CT) [12], which contain 78,989, 66,724 and 306,591 concepts, respectively. The LargeBio track consists of six tasks corresponding to the different sizes of input ontologies (small fragments/whole ontology of FMA and NCI and small/large fragments of SNOMED-CT; see Table. 2). The Unified Medical Language System (UMLS) [5] has been used as the basis to produce the reference alignments [7].

7.2. Preselected ontologies

According to the OAEI 2016 campaign, AML [22] and LogMapBio [31] are the best BK-based ontology matching systems. To establish a fair comparison with these systems, our evaluation employs the same set of preselected ontologies as follows:

AML-Ontologies: Three ontologies are preselected for AML: UBERON, DOID and MeSH³. AML makes a dynamic selection from these ontologies using the *Mapping Gain* measure [23].

LogMapBio-Ontologies: In OAEI2016, LogMapBio considered the NCBO BioPortal as the set of preselected

ontologies. LogMapBio selected 10 ontologies for each matching task. For our evaluation, we considered the combination of all the ontologies selected by LogMapBio as the preselected ontologies in order to establish a fair final result comparison. The combination yields 21 ontologies. However, YAM++ could not parse three of those ontologies. Indeed, these ontologies require importing external ontologies, a process which is not managed by YAM++. Thus, we ended up using 18 (out of the 21) ontologies for our comparison with LogMapBio. These ontologies are listed in Table 3 with their NCBO BioPortal acronyms⁴.

For each matching task, we name **BBK1** the background knowledge resource built from AML-Ontologies and **BBK2** the one built from LogMapBio-Ontologies. Building the BK is performed according to the process described in Section 3 with YAM++ as a matcher. The extracted mappings, the candidate mappings, as well as the source code are openly available⁵.

7.3. Tools and resources

YAM++. We used YAM++ to generate all the required alignments for our experiments. YAM++ is an ontology matching system previously developed by our team at LIRMM⁶ [41]; it does not rely on a specialized BK to match biomedical ontologies. It is considered as one of the state-of-the-art ontology matching systems, and was the top ranked system in OAEI 2013. YAM++ combines several syntactic, lexical and structural similarity measures.

OBO x-refs. In addition to the mappings generated by YAM++, we also extracted cross-reference properties from the preselected ontologies when available (i.e., from the preselected ontologies present in the OBO Foundry). As previously pointed out (see Section 3.2), these cross-references may be considered as manually curated mappings. Therefore, we added them to the extracted mappings and assigned them a score of 1 when computing candidate mapping scores.

⁴These ontologies are accessible on NCBO BioPortal with the link <https://bioportal.bioontology.org/ontologies/ontologyAcronym>

⁵<https://github.com/AminaANNANE/BK-based-matching>

⁶<http://www.lirmm.fr/yam-plus-plus>

²<http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/>

³MeSH is used as lexicon [21]

Table 3: LogMapBio-Ontologies (ontologies tagged by * could not be parsed by YAM++).

N°	Ontology acronym	Number of concepts
1	BIRNLEX	3,580
2	BTO	5,902
3	CCONT	19,991
4	CL*	2,352
5	CLO	40,884
6	CSEO	20,085
7	DDO*	6,444
8	DINTO*	28,178
9	DOID	12,432
10	EFO	19,909
11	EHDAA2	2,772
12	GALEN	23,141
13	HP	15,804
14	MA	3,257
15	ONTOAD	5,899
16	RCTV2	88,854
17	SYN	14,462
18	UBERON	19,761
19	VHOG	1,185
20	XAO	1,621
21	ZFA	3,168

Neo4j. Technically, the mapping filtering step produces two files: (i) an OWL file containing all selected concepts with their labels and ontology source, and (ii) a CSV file containing all mappings in format (URI source, URI ontology source, URI target, URI ontology target, score, manualMapping). *manualMapping* is a boolean property that takes "true" or "false" as value. The OWL file is used for anchoring the target ontology to the built BK. The mapping file is stored as a graph database using Neo4j⁷, where each node is unique and described by its URI and ontology source. A graph database facilitates the derivation step. Indeed, with relational databases, one has to implement an algorithm and perform several queries to find all paths between a given source and target concepts. Instead, with a graph database, which is intrinsically designed to work with paths within graphs, a single simple query is sufficient.

Weka. Weka⁸ is an open source software that includes a collection of machine-learning algorithms for data mining tasks. We used the RandomForest algorithm included in Weka [26].

Machine specifications. We run our experiments on an HP ZBook computer that has an Intel Core i7-4910MQ processor, 2.90 GHz of clock, 32 GB of RAM, and a 64-bit Operating System (Windows 8.1 pro).

8. Experimental evaluation

In this section, we evaluate our BK-based ontology matching approach through several experiments. We organize the evaluation in five sections. Each section is introduced with an assumption that we try to validate through experiments.

8.1. Assumption 1: *Our BK selection method builds a smaller-size BK than the preselected ontologies*

As discussed previously, our BK selection approach does not return a set of ontologies. Instead, it builds a BK that combines concepts selected from the various preselected ontologies. To verify the assumption of this section, we compare the size (i.e., the number of concepts) of our built BK with that of the preselected ontologies (i.e., AML-Ontologies and LogMapBio-Ontologies).

In Table 4, for each matching task, we present the size of the built BK (BBK1 or BBK2) in number of concepts. Furthermore, we compute a percentage by dividing the size of the built BK by the size of the preselected ontologies. BBK1 is built from three ontologies, which have a global size of 297,031 concepts while BBK2 is built from 18 ontologies, which have a global size of 302,707 concepts. For instance, the size of Task 1 BBK1 is 6,809; dividing 6,809 by 297,031 gives a percentage of 2%, which means that the BBK1 size represents only 2% of the preselected ontologies size.

The results reported in Table 4 validate Assumption 1. Indeed, for all matching tasks, the size of the built BK is much smaller than the size of the preselected ontologies. The percentage varies from one task to another with respect to the size of the ontologies to align. Tasks 2 and 6 share exactly the same built BK because they have the same source ontology (see Table 2); this shows that, when matching the source ontology with several target ontologies, the BK selection step may be performed once, and the built BK can be reused for each target ontology.

Table 4: Size comparisons: built BK vs. preselected ontologies.

Task	BBK1 size		BBK2 size	
Anatomy	3,173	1%	11,090	4%
Task 1	6,809	2%	18,104	5%
Task 2	46,280	15%	48,521	16%
Task 3	13,036	4%	27,465	8%
Task 4	16,251	5%	34,626	10%
Task 5	12,895	4%	36,456	12%
Task 6	46,280	15%	48,521	16%

8.2. Assumption 2: *Deriving mappings across several intermediate concepts generates more correct mappings than deriving across one intermediate concept.*

Deriving candidate mappings is performed by searching paths between source and target concepts. Each path contains a number of intermediate concepts belonging to

⁷<https://neo4j.com/>

⁸<https://www.cs.waikato.ac.nz/ml/weka/>

the preselected ontologies. For instance, with three preselected ontologies, we may derive mappings with paths that contain one intermediate concept, two intermediate concepts and three intermediate concepts. In these experiments, we derived mappings with a maximum of three intermediate concepts. Indeed, according to our experiments in [3], deriving mappings with more than three intermediate concepts generates much more incorrect mappings than correct ones.

To verify that deriving mappings across several intermediate concepts generates more correct mappings, we derived all possible candidate mappings between the ontologies to align for each matching task. We then computed (1) A: the number of correct mappings derived with only paths containing one intermediate concept; (2) B: the number of correct mappings derived with paths containing one, two and three intermediate concepts; (3) Gain: the percentage of gain when using paths with several intermediate concepts ($\text{Gain} = \frac{B-A}{A}$). The results are reported in Table 5.

As we can see, Assumption 2 is validated. Indeed, for each matching task, the derivation across several intermediate concepts generates more correct mappings, with a gain of up to 7%, compared to deriving mappings with only one intermediate concept. Tasks 1 and 2 have the same values, since they have the same reference alignment. The same is true for Tasks 3 and 4, Tasks 5 and 6. Note that deriving mappings with only one intermediate concept is comparable to deriving with each BK ontology independently (i.e, separately from the other BK ontologies, composing only two anchoring mappings related to the same BK ontology), which is the method adopted by almost all related works [22, 30, 27, 46]. Instead, thanks to the mapping extraction task, our approach combines all preselected ontologies.

For Anatomy, the gain is not significant. This may be explained by the use of UBERON, which is an integrative multi-species anatomy ontology. Indeed, UBERON, employed as the only BK ontology, allows to identify more than 80% of Anatomy reference alignment mappings.

When deriving mappings across several intermediate concepts, we may notice that the number of correct mappings derived with BBK1 is comparable to BBK2 for Tasks 1, 2, 3 and 4. However, for Tasks 5 and 6, the gap is larger: 10,315 correct mappings are derived with BBK2 while only 5,091 correct mappings are derived with BBK1. This shows that BBK2 is more effective than BBK1 for these tasks.

8.3. Assumption 3: Our rule-based and ML-based mapping-selection methods are effective

Exploiting background knowledge resources in ontology matching generates more correct and incorrect mappings (as previously discussed), selecting the most relevant mappings is a crucial step. We proposed and described two mapping selection methods in Section 3. Here we evaluate these methods to validate Assumption 3.

Table 5: Evaluation of derivation effectiveness using several intermediate concepts.

BK	Task	A	B	Gain
BBK1	Anatomy	1,403	1,405	0.1%
	Task 1 & Task 2	1,938	2,054	6.0%
	Task 3 & Task 4	2,043	2,158	5.6%
	Task 5 & Task 6	4,789	5,091	6.3%
BBK2	Anatomy	1,411	1,420	0.6%
	Task 1 & Task 2	2,369	2,442	3.1%
	Task 3 & Task 4	2,511	2,685	6.9%
	Task 5 & Task 6	9,871	10,315	4.5%

Table 6: Correct and incorrect mappings in the baseline.

Task	BBK1		BBK2	
	Correct	Incorrect	Correct	Incorrect
Anatomy	1,416	135	1,420	157
Task 1	2,463	156	2,497	248
Task 2	2,463	772	2,497	1,402
Task 3	4,586	423	4,613	653
Task 4	4,586	1,017	4,613	1,661
Task 5	12,415	4,135	13,184	4,877
Task 6	12,415	6,345	13,184	7,948

To carry out our experiments, we employed the following mapping selection methods:

1. **Baseline.** This is the simplest method. It keeps all candidate mappings that have been derived without any selection. In Table 6, we present the number of correct and incorrect mappings in the baseline.
2. **Rule-based selection.** This method implements the rules described in Section 5.1 to select the final mappings.
3. **ML-based selection.** To evaluate ML mapping selection, we implemented two strategies which use the same ML algorithm (RandomForest) and the same attributes to describe candidate mappings but for which we generate training data differently. Note that, in both strategies, test data (candidate mappings to classify) and training data are completely distinct.
 - (a) **Cross validation.** This strategy is often used to evaluate the performance of the ML algorithm and the attributes when using training data objects that are similar to the objects to classify. The process for a given matching task is as follows: (1) we randomly subdivide the set of candidate mappings of this task into two equal subsets. (2) Based on the reference alignment, we annotate the candidate mappings of the first subset with *true* or *false*. (3) We use the annotated subset as training data to learn a classifier. (4) We then classify the candidate mappings of the second subset with the resulting classifier. (5) We interchange the two

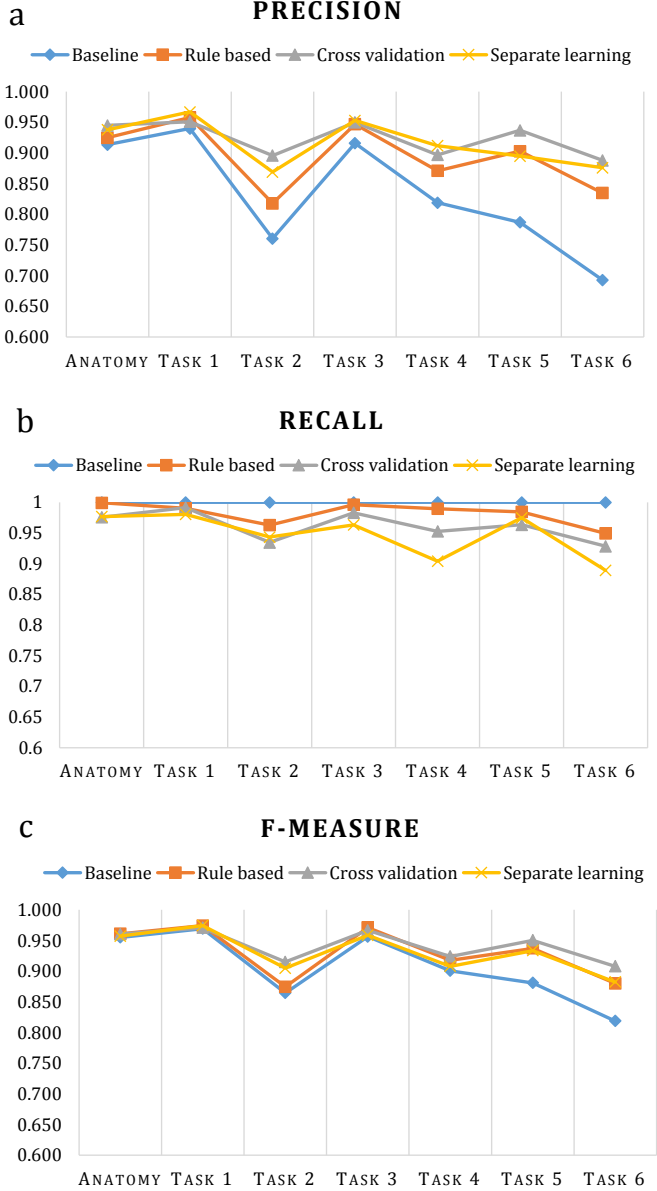


Figure 9: Evaluation of mapping selection methods using BBK1.

subsets such that we annotate the second subset and classify the candidate of the first one. (6) Finally, we combine the two classification results (taking all the candidate mappings classified as *true*) to obtain the final alignment.

- (b) **Separate learning.** Here, we generate the training data for a given matching task using the ontologies and reference alignments of other tasks. For LargeBio, we adopt a *leave-one-out* strategy. For each task, we generate the training data using other same-size tasks. For instance: there are three large fragments tasks (Tasks 2, 4 and 6), to classify Task 2 candidate mappings, we use the ontologies and reference alignments of Tasks 4 and 6 to generate the training data, according to the process illustrated in Figure 7. For Anatomy, we generated

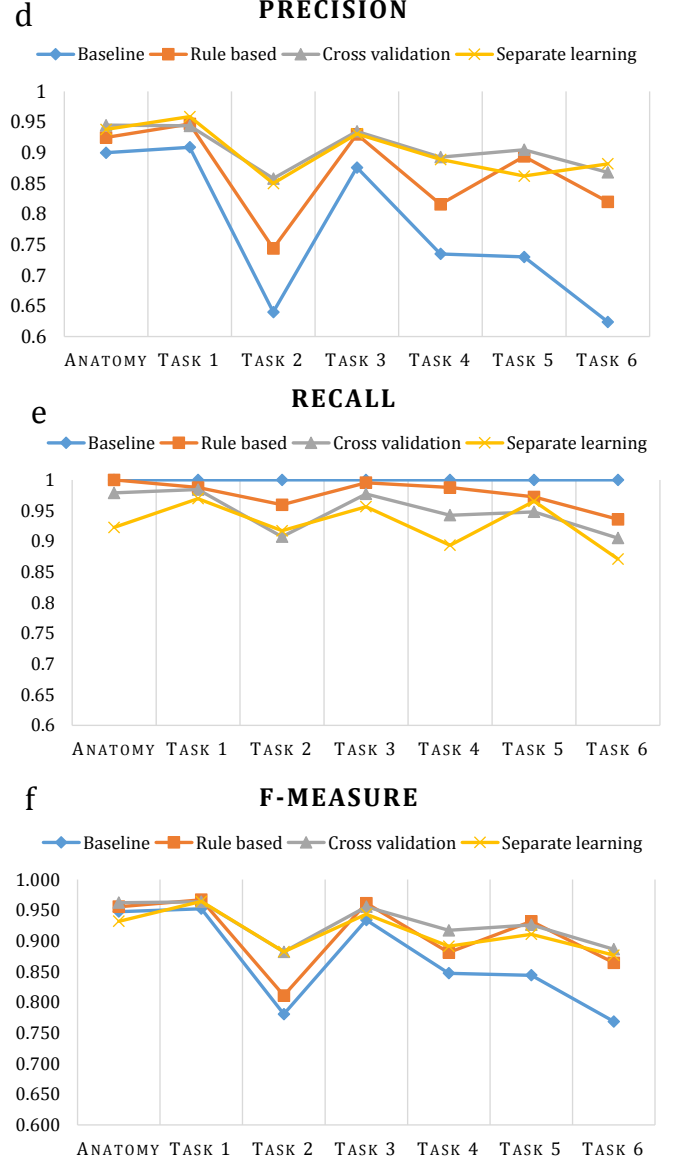


Figure 10: Evaluation of mapping selection methods using BBK2.

the training data with Tasks 1, 3 and 5.

In this section, to evaluate the performance of our selection methods fairly, we compute the recall with respect to the number of derivable correct mappings but not to the number of mappings in the reference alignment. Indeed, if some correct mappings are not available in the set of candidate mappings, we cannot blame the selection method for not having returned them.

$$Recall = \frac{TP}{TPG}$$

Where TP is the number of the correct mappings returned by a given selection method, and TPG is the number of all correct mappings derivable with the built BK.

Figures 9 and 10 present the results of our experiments for each matching task exploiting respectively BBK1 and BBK2. In particular, we show the precision, recall and

F-measure of the produced alignments to observe the behavior of each mapping selection method.

Precision: As we can see in Figures 9 (a) and 10 (d), the baseline’s precision for small size tasks (Task 1, Task 3 and Anatomy) is comparable to that of other selection methods. However, for larger size tasks (Tasks 2, 4, 5 and 6), the precision is low, especially for Tasks 2 and 6. Even if the precision curves display the same trend in Figures 9 (a) and 10 (d), the scores in Figure 10 (d) are lower than those in Figure 9 (a). This may be explained by the fact that BBK2 is built from a larger number of preselected ontologies than BBK1 (18 vs. 3 ontologies). Hence, BBK2 generates more correct (see Table 5) and incorrect mappings, which decreases precision. The ML-based selection methods consistently yields higher precision than the rule-based selection method, with an average of 0.915 for cross-validation and 0.909 for separate learning (vs. 0.881 for the rule-based selection). The largest gap is observed in Task 2. This is due to the fact that the NCI Thesaurus includes a small branch on mouse anatomy (in addition to the human anatomy branch). Using the cross-references extracted from UBERON (considered as manual mappings) and the selection rule number 2 (see Section 3), the rule-based selection method returns mappings between human and mouse anatomy. However, the UMLS, the source from which the reference alignment is extracted, is focused only on human health, and does not include mappings between the NCI mouse anatomy branch and MA; therefore, these mappings are considered as incorrect, which affects precision.

Recall: The baseline always shows a recall of 1, because we computed a customized recall as described above (see Figures 9 (b) and 10 (e)). Our selection methods yield a high recall in all matching tasks. The rule-based mapping selection method obtained the best recall scores, with an average of 0.979, while the cross-validation and separate-learning methods have a recall average of 0.955 and 0.938, respectively. The difference between the rule-based and separate-learning selection is significant in Task 4, while the gap is smaller with cross-validation. This may be explained by the low precision of the baseline alignment of Tasks 2 and 6. This affects the learned classifier. Indeed, the baseline alignments of Tasks 2 and 6 are the training data of Task 4. Training data contains many *false* candidate mappings increase the probability of classifying a given candidate mapping as *false*, which, in turn, decreases recall.

F-measure: We present the F-measure values in Figures 9 (c) and 10 (f). The cross-validation method yielded the best F-measure scores with an average of 0.942 when using BBK1 and of 0.928 when using BBK2. These results demonstrate that the ML technique with the proposed attributes and similar data training is effective for mapping selection. ML-based mapping selection is therefore particularly well-suited for complementing an existing partial alignment between two ontologies (the partial alignment may be used to generate the training data) [34, 38], or for

matching new ontology versions when an alignment between the old ontology versions already exist. Indeed, the training data may be generated with the existing alignment.

The separate-learning method produced high F-measure scores as well, close to the cross-validation method’s scores, with an F-measure average of 0.931 and 0.914 when using BBK1 and BBK2, respectively. These results are more interesting. They represent a concrete case where we may reuse existing alignments within the same domain to learn an effective classifier. Note that, we generated the training data for the Anatomy (that has a gold standard reference alignment) using alignments extracted automatically from UMLS (Tasks 1, 3 and 5 reference alignments), and the selection results are promising.

The rule-based method provides results with an average of 0.931 and 0.910 when using BBK1 and BBK2, respectively. It obtained the best F-measure values for the small tasks (i.e., Anatomy, Tasks 1 and 3). However, its performance decreases (i.e., achieves lower precision) for large tasks, compared to ML-based selection, which is more stable.

The results of the ML-based and rule-based mapping selection methods are comparable in terms of F-measure scores. However, the ML-based selection promotes precision, while the rule-based selection promotes recall.

Rule-based mapping selection is simple and efficient but static. Indeed, although each ontology-matching task has its own specificities (for instance, the best threshold value varies from one task to another), the same rules apply all the time. ML-based mapping selection is time consuming and requires aligned ontologies to generate training data. However, it dynamically learns a customized classifier that combines multiple selection attributes (27 in our case). Mappings that are manually created or validated stored in platforms such as YAM++ online [4]⁹, NCBO BioPortal or resources such as OBO ontologies may be used to generate training data.

Based on our experiment results, we can validate Assumption 3. Our selection methods are effective: they significantly improve baseline precision and consistently keep high recall.

8.4. Assumption 4: *The use of ontologies as background knowledge has a computation time cost and our built BK reduces this cost*

In this section, we describe our computation time evaluation. We start by analyzing and discussing the time necessary to perform each step of our approach. We then present the efficiency gain obtained.

8.4.1. Computation time evaluation of our approach: step by step

Figures 11 and 12 present the time, in minutes, required for the different steps in our approach.

⁹<http://yamplusplus.lirmm.fr/>

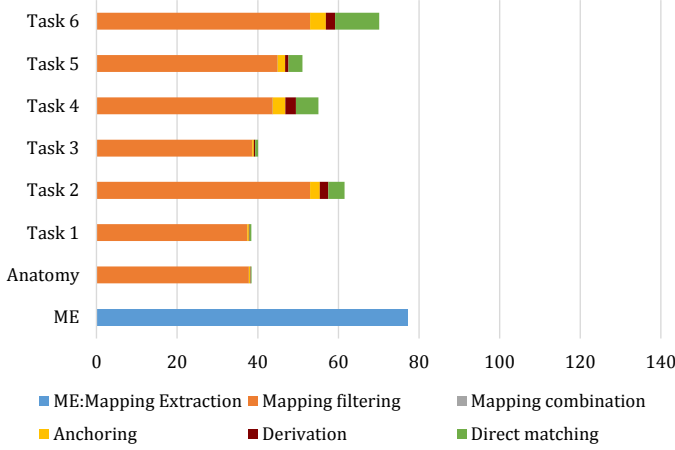


Figure 11: BK selection and BK use computation time in minutes (BBK1).

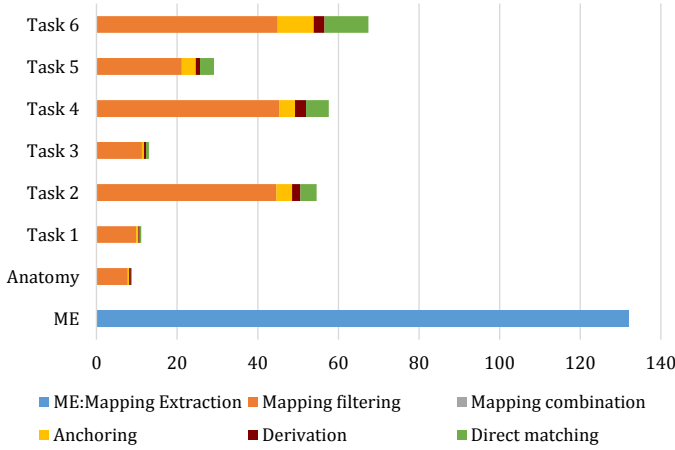


Figure 12: BK selection and BK use computation time in minutes (BBK2).

BK selection

The BK selection step includes three tasks: mapping extraction, mapping filtering and mapping combination.

Mapping extraction is the costliest task in terms of computation time, especially when using a large number of preselected ontologies, as is the case for BBK2 (18 ontologies). Indeed, extracting the mappings from the AML-Ontologies and LogMapBio-Ontologies took 77 and 132 minutes, respectively. Fortunately, this task is performed independently of matching tasks. In fact, for a given set (or repository) of preselected ontologies, the mapping extraction task is performed only once whereas its output (the set of alignments) is reused for any matching task. Therefore, we report the computation time of the mapping extraction process once for all matching tasks in Figures 11 and 12.

BBK1 is built from three ontologies. However, the time necessary for extracting the mappings from the three ontologies is 58% the time necessary for performing the same process from 18 ontologies. This may be explained by the fact that, in terms of computation time, matching a large preselected ontology such as MeSH is equivalent to match-

ing several small ontologies.

The high computation time cost of the mapping extraction step is justified by the fact that the mapping derivation across several intermediate concepts generates more correct mappings, as demonstrated in Section 8.2.

Mapping filtering is the second costliest process. It includes two tasks: (i) matching the source ontology to the preselected ontologies and (ii) selecting the mappings related to the source ontology. The first task is time-consuming, especially when dealing with large scale ontologies such as MeSH. Indeed, it is surprising to notice that matching the source ontology to 3 preselected ontologies takes more time than matching it to 18 preselected ontologies (see Figures 11 and 12). MeSH contains 265,414 concepts, and each concept is described with multiple labels. Therefore, YAM++ takes long time to match MeSH with the source ontology, particularly when the latter is a large-scale one. The second task takes only few seconds in all cases.

Mapping combination is performed with Neo4j allowing us to merge the same nodes of different mappings. It takes less than 2 seconds in all cases.

BK use

This step consists in exploiting the built BK to derive candidate mappings (anchoring and derivation). It takes much less time than the BK selection step. The size of the target ontology is larger than that of the source ontology (see Table 2), however, we notice that anchoring the target ontology takes less time than matching the source ontology to the preselected ontologies in the mapping filtering step. This may be explained by the fact that the target ontology is anchored only to the reduced-size built BK.

The derivation task is performed with Neo4j. It takes less than one minute for small matching tasks and up to three minutes for large ones.

Final mapping selection

The computation time for the rule-based mapping selection method is less than two seconds in all cases. However, when using ML-based selection, computation time is much longer. This is mainly due to the generation of training data. For example, in our evaluation, we used Tasks 1 and 3 to generate the training data for classifying Task 5 candidate mappings. Hence, the time necessary to generate the training data in this case is the time necessary for BK selection and BK use of Tasks 1 and 3. Note that the training data for Task 5 may be generated with Task 1 only. Furthermore, the learned classifier is reusable in the same domain. Indeed, we tried to classify Task 4 candidate mappings derived from BBK2 with the classifier trained with Tasks 2 and 6 candidate mappings derived from BBK1. We obtained almost the same results as those obtained with the classifier trained with candidate mappings derived from BBK2. Hence, spending time to learn one classifier for a given domain is acceptable, because it can be reused for different matching tasks. Learning a classifier and classifying candidate mappings are less time consuming and take only few seconds.

Comparing the time necessary for direct matching to that required for BK-based matching shows that the use of BK in ontology matching is time-consuming.

8.4.2. Efficiency gain with the built BK

Our approach reduces the computation time of the BK-based matching process, especially that of BK selection and anchoring as explained in Section 6. In our evaluation we used the same matcher (i.e., YAM++) for BK selection and anchoring. Hence, we are in **Case 1** that reuses the BK selection alignments as anchoring alignments. We compare T and T' computed according to the formulas (1) and (2) introduced in Section 6, respectively.

In Figures 13 and 14, we present: (i) T : the time necessary for matching the source and target ontologies to the preselected ontologies in the traditional approach. We ignore α because it has a small value and varies from one work to another as explained in Section 6; (ii) T' : the time necessary for mapping filtering, mapping combination and anchoring the target ontology to the built BK in our approach, and (iii) the percentage ratio comparing the two. This ratio is computed by dividing T' by T .

In all cases, our approach is more efficient than the traditional approach (i.e., $T' < T$). The gain is between 42% (for Task 2 with BBK1) and 60% (for Task 4 with BBK2). These results are expected since T and T' have a common part: matching the source ontology to the preselected ontologies. However, matching the target ontology to the preselected ontologies takes more time comparing to matching the target ontology to the BBK. For instance, in all tasks, matching the target ontology to BBK1 takes less than four minutes, while matching the target ontology to the large ontology MeSH always takes about 30 minutes.

With YAM++, the average time to match an ontology of LargeBio or Anatomy to: (i) one of the 18 ontologies listed in Table 3 is 2.8(min), (ii) the BK built from the 18 ontologies is 5.5(min). We may use these values to check our intuition about the efficiency gain in **Case 2**. When selecting only one ontology as background knowledge from the 18 ontologies, the difference between T and T' in Case 2 is computed as follows.

$T - T' = E + (2.8 + 2.8) - 5.5 = E + 0.1(min)$ where E is the value of the formula (3) that is the difference between T and T' in Case 1. As we can see in Figure 13 and 14, E is always positive. Hence, $E + 0.1(min)$ is positive.

In Case 2, the efficiency gain becomes more significant as the number of selected ontologies increases. For instance, with two ontologies as BK, the difference becomes: $T - T' = E + (2.8 * 2 + 2.8 * 2) - 5.5 = E + 5.7(min)$

Based on the obtained results, we conclude that our BK selection approach builds an efficient BK, which validates Assumption 4. Indeed, the built BK reduces the BK selection and anchoring computation-time comparing to the traditional approach.

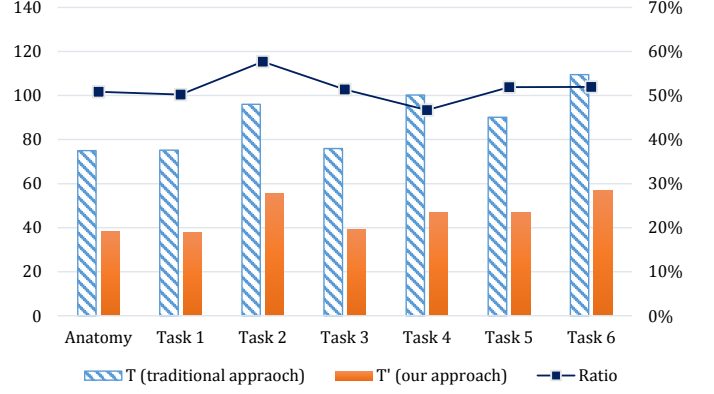


Figure 13: Efficiency gain with BBK1.

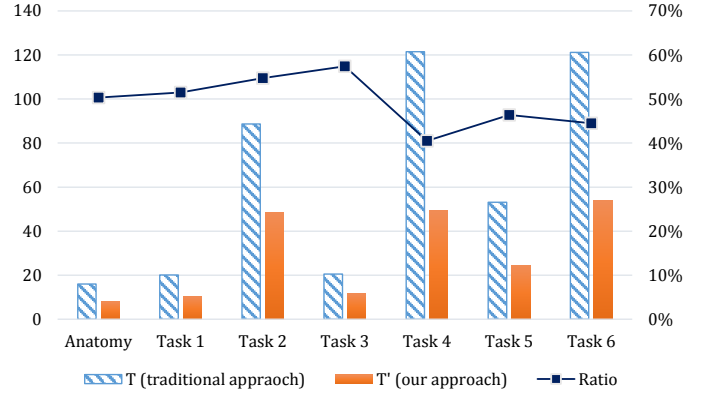


Figure 14: Efficiency gain with BBK2.

8.5. Assumption 5: The small size of the built BK does not affect its effectiveness

According to OAEI campaigns [17], AML and LogMapBio are the best systems using ontologies as background knowledge. To verify the effectiveness of the built BK, we compare our results to theirs. For a fair comparison, (i) our evaluation uses the ontologies that were preselected for these systems in the OAEI 2016 campaign, (ii) only our rule-based selection results are compared since the OAEI rules prohibit training on OAEI datasets¹⁰, and (iii) we repaired the alignments generated by our approach with the LogMap's ontology repair module (LogMap-Repair) [32], which is available as a self-contained software component¹¹. Indeed, AML and LogMapBio use logical repair strategies to ensure the coherence of their alignments.

The aim of this comparison is to evaluate the performance of our approach regarding the best results obtained using the same preselected ontologies. Thus, if we obtain comparable results, we can conclude that the reduced size of the built BK does not affect its effectiveness.

Note that, in this section, we compute the recall against the reference alignment, as described in Section 2.2.

In Table 7, we present the difference between the F-measure values of the repaired alignments and those of

¹⁰<http://oaei.ontologymatching.org/doc/oaei-rules.2.html>

¹¹<https://code.google.com/archive/p/logmap-matcher/downloads>

the original ones. Generally, repairing the alignments with LogMap-Repair has a positive impact on the F-measure values. This impact is more significant when using BBK2, especially for Task 2 and 4. This may be explained by the fact that BBK2 generates more incorrect mappings than BBK1 (see Table 6), hence more inconsistent mappings.

Table 7: Repairing gain with LogMap-Repair.

Task	Gain (BBK1)	Gain (BBK2)
Anatomy	0.003	0.003
Task 1	-0.001	0.001
Task 2	0.007	0.034
Task 3	0.001	0.003
Task 4	0.002	0.015
Task 5	0.001	0.001
Task 6	0.004	0.008

In Figures 15 and 16, we present the F-measure scores of the alignments returned by:

- AML when our approach exploits BBK1.
- LogMapBio when our approach exploits BBK2. The results of AML and LogMapBio are those reported in the OAEI 2016 campaign.
- Our approach with the rule-based mapping selection method, and repaired with the LogMap-Repair module.
- YAM++ (direct-matching results)

Our results vs. YAM++ results

Our approach significantly improved the F-measure values of the direct matching performed with YAM++, mainly by increasing recall. For instance, for Task 5 with BBK2, our approach increased F-measure score from 0.740 to 0.814. This may be explained by the effectiveness of our built BK, which generated more correct mappings (high recall), and of the mapping selection methods, which insured high precision too. These results legitimate the current trend of using BK resources to enhance ontology matching.

Our results vs. state-of-the-art results

Our approach yields slightly higher F-measure than AML in three tasks (Tasks 1, 3 and 4) and very close results in the other tasks (see Figure 15). We may state that our results are comparable to AML’s results when exploiting the three preselected ontologies. Indeed, we obtained almost the same F-measure average (0.843).

Our approach outperforms LogMapBio, yielding superior results in all tasks except Task 2 (see Figure 16). This may be due to the derivation across several intermediate concepts, which increases our results’ recall and F-measure. Indeed, LogMapBio composes only two mappings related to the same BK ontology at a time. Direct matching with YAM++ yields the best F-measure for Task 2. This may be explained by the significant loss in

precision due to the use of the UBERON ontology, as explained in Section 8.3.

The obtained results validate Assumption 5. Indeed, the built BK enhances the quality (F-measure) of the direct-matching alignment, and provides results competitive to the state-of-the-art results.

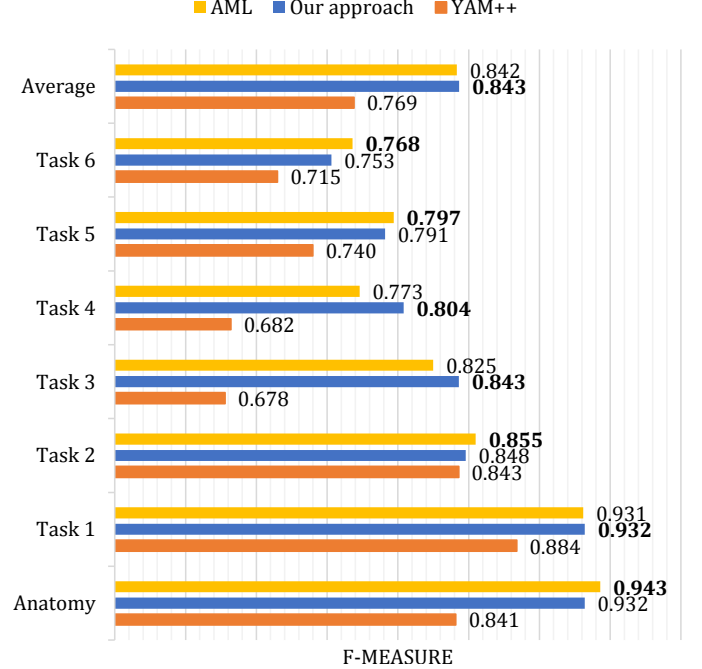


Figure 15: Final alignment F-measures using BBK1.

9. Discussion and limitations

As we have mentioned in Section 3.3, to improve our approach efficiency, we consider the smallest ontology as the source ontology. We tried to check whether exchanging the ontology positions (i.e., source ontology becomes target ontology and vice versa) has an impact on the results in terms of Precision, Recall and F-measure.

Theoretically, this may happen when we reuse the BK selection alignments as anchoring alignments (i.e., **Case 1** in Section 6). Indeed, the source ontology is anchored to the built BK using the syntactic and structural content of the preselected ontologies, while the target ontology is anchored to the built BK using only the syntactic information of the selected concepts (i.e., labels). In Figure 17, we illustrate this case with an example.

Let O_1 , O_2 be two ontologies to align; O_p a preselected ontology; e_1 , e_p and e_2 three concepts belonging to O_1 , O_p and O_2 , respectively. We suppose that matching e_1 to e_p requires structural techniques that exploit the hierarchy of O_1 and O_p , while e_2 can be matched to e_p only with syntactic or lexical techniques.

When O_2 is the source ontology, the mapping between e_2 and e_1 cannot be derived, since the structural information of the ontology O_p is not kept in the built BK.

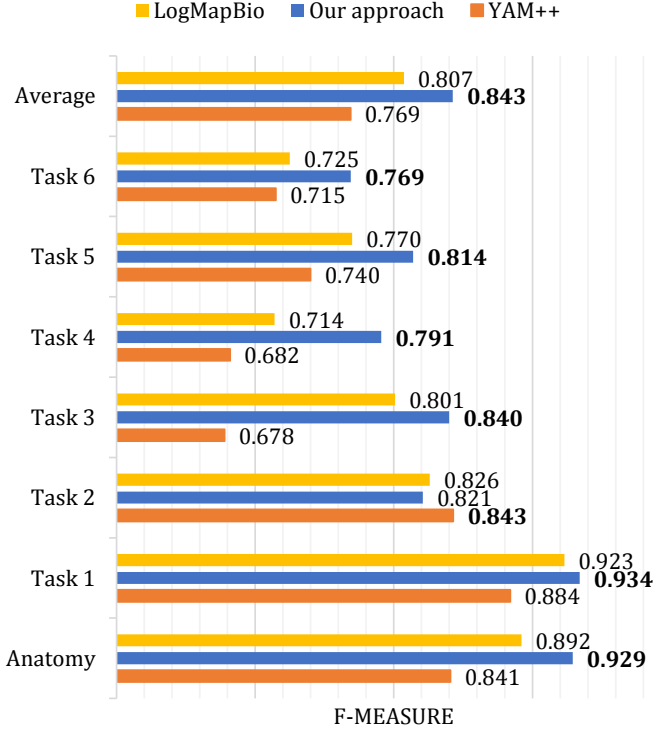


Figure 16: Final alignment F-measures using BBK2.

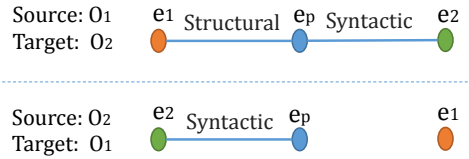


Figure 17: Example of exchanging source and target ontologies.

We performed the same experiments as those presented in Section 8.5 exchanging the order of the source and target ontologies. The results did not change. This may be explained by the fact that the case described above is rare, at least in the used benchmarks. Indeed, discovering mappings based on the structural information is difficult as different ontologies can have different models of the same concept [45]. Usually, the structural information is mainly used to endorse mappings found thanks to syntactic or lexical techniques [41, 14].

In this paper, we focus only on deriving equivalent mappings. To extract mappings of various relations, the BK selection method should be customized. For instance, to derive mappings with isA relation, for each selected concept, the BK selection method should extract the first level or the nth level—where n is a parameter—of its descendants and ascendants. Thus, the built BK concepts will be interconnected via isA and equivalence relations, which will enable deriving mappings with these two relations.

The effectiveness of the built BK to improve the direct matching alignment depends on two factors: (i) the initial set of preselected ontologies. Indeed, if the preselected ontologies are not semantically rich (e.g., concepts with no synonyms or definitions), and there is no overlap be-

tween them and the ontologies to align, our approach, as all the BK-based matching approaches, will not improve the direct matching alignment; and (ii) the quality of the alignments used to build and exploit the BK. Hence, the matcher that generated these alignments (YAM++ in our case). We believe that the more effective is the matcher, the higher quality will have the alignments generated by our approach.

We have evaluated our approach on two OAEI benchmarks. Our choice was motivated by the fact that only for these tracks, the state-of-the-art systems use ontologies as background knowledge. Hence, evaluating on these benchmarks allows us to compare our results to the state-of-the-art ones. However, these benchmarks include ontologies of one domain, the biomedical domain, which may be considered as a limitation of our evaluation. Nevertheless, the biomedical domain is suitable for evaluating the BK-based matching approaches for two reasons: (i) the vocabulary of ontologies is complex and specialized, which limits the effectiveness of syntactic similarity measures and generic lexical resources such as WordNet [23]; (ii) there are many biomedical ontologies with overlapping fragments, which can be exploited as background knowledge.

10. Related work

In this section, we review related work on the three main topics of this article: (i) BK selection, (ii) BK-based ontology matching, and (iii) ontology matching using ML.

10.1. BK selection

Several approaches have been proposed to select BK for a given matching task. Hartung et al. [27] have proposed a measure to rate the effectiveness of a given preselected ontology for an ontology matching task. This measure is based on the overlap between the ontologies to align and the preselected ontologies. The authors hypothesize that a larger overlap means a better BK ontology. Another effectiveness measure, called mapping gain, was proposed in [23]. It measures the number of new mappings in an alignment A derived using a given BK ontology, comparing to another alignment B generated without this BK ontology. The methodology comprises two stages: ranking and selection. In the first stage, candidate ontologies are identified. In the second one, the candidate ontologies are reevaluated and ranked, taking into account already selected ontologies. This is the method used by the AML ontology matching system. The two measures described previously are computed using the alignments generated by matching the ontologies to align to all the preselected ontologies. In [37], the authors used the simple number of anchors between the ontologies to align and the preselected ontologies as a measure to select the ontologies to be exploited as background knowledge.

Relying on the information retrieval technique, Quix et al. [46] selected a set of ontologies as background knowl-

edge. Their approach consisted in using the lexical information (labels, definitions, etc.) of ontologies present in a given local repository to index them. Ontology structural information has been exploited too. Indeed, the approach prioritizes ontologies that have a rich class hierarchy, because they allow more relationships to be inferred. Many ontologies can be returned by the local search engine for each query. To select the top ontologies, a customized ranking is necessary. The authors therefore proposed a measure to prioritize the ontologies closest to both ontologies to align. A similar approach was suggested in [54].

Chen et al. [9] have investigated the use of the NCBO BioPortal [43] as a dynamic mediating ontology provider. The authors have suggested the use of a matcher to produce a direct alignment between the source and the target ontologies, and the subsequent extraction of concept labels for this alignment. For each label, the algorithm searches the NCBO BioPortal for ontologies that contain this label. The algorithm stops if the number of identified BK ontologies does not change after a specified number n of calls to the NCBO BioPortal, or when there are no more labels to check. Finally, the top ontologies are selected according to their number of labels and synonyms. This is the method used by the LogMapBio ontology matching system.

The novelty of our approach lies in the fact that it considers concepts as units of selection rather than complete ontologies. Indeed, our approach selects, from each preselected ontology, only those concepts that are related to the source ontology. Hence, it significantly reduces the research space for the target ontology, whereas, in [23, 27], the target ontology is matched to all the preselected ontologies.

10.2. BK-based ontology matching

Several ontology matching systems exploit knowledge resources to enhance their results. For instance, among the 50 systems that participated in Anatomy track, 35 systems used an external knowledge resource [14]. Two main techniques are implemented to exploit these resources. The first one consists in enriching the concept labels of the ontologies to align with synonyms offered by the BK. This technique is especially relevant when using a lexical resource, such as WordNet, as background knowledge [36]. The second technique is based on mapping composition. It consists in matching the ontologies to align to one or several BK ontologies and to compose the resulted mappings to identify new ones between the source and target ontologies.

Some works use each BK ontology independently of the others: they compose mappings only related to the same BK ontology [46, 27]. This technique is employed by the GOMMA [25], AML and LogMapBio ontology matching systems, while other works have investigated the possibility of identifying new mappings across several BK ontologies. In addition to anchoring, BK ontologies are matched between each other [29, 50, 37]. In this case, the derivation is performed either with a debugging component [29],

or with a process known as *path-driven* derivation, which composes mappings linking different BK ontologies [50, 37].

In our approach, we adopted the path-driven derivation. The difference is that we do not deal with whole ontologies in the derivation step; instead, we exploit the customized built BK as a graph (or network) of concepts interconnected via mappings. This improves overall efficiency. Furthermore, although the reduced-size of the built BK, it allows us to derive mappings across one or several intermediate ontologies. Note that, in this paper we only deal with deriving equivalence mappings.

10.3. Supervised machine learning for ontology matching

Though the general workflow of a supervised machine learning approach remains generic, it requires the definition of two key parameters: the attributes that describe the training data and test data objects; and the manner in which to obtain or generate the training data (classified objects). Syntactic, structural and lexical similarity measures are used as attributes to describe a candidate mapping. However, the generation of training data varies from one work to another.

In [53], the authors proposed to use manually-produced mappings as training data. This technique was criticized because it is based on the cognitive (memory and decision-making) abilities of the users charged with manually creating the mappings. In addition, it is fastidious, and may not provide enough data to learn a good classifier [13]. Other tools, such as APFEL [16], generate mappings automatically, and ask users to validate them. Both correct and incorrect mappings are used as training data. Despite the drawbacks of user-generated training data, their advantage is that user preferences are inherently captured in the mappings. For instance, in Task 2, based on user preferences, mappings between the NCI mouse anatomy concepts and FMA human anatomy concepts may be considered correct or incorrect. Another solution was adopted in [47], which considers mappings generated automatically as training data. This technique generates a larger training data set than the previous technique. However, it does not take user preferences into account. Moreover, it may consider incorrect mappings as correct, which affects the learned classifier.

Our approach combines the two approaches described above: it automatically generates candidate mappings, which provides enough training data. Then, it annotates the generated candidate mappings with a reference alignment which, in turn, ensures high training data quality.

11. Conclusion

In this paper, we make a contribution to the BK-based ontology matching by designing a complete approach that dynamically builds a BK with concepts chosen from a set of preselected ontologies. Building that BK eliminates the

need to deal with complete BK ontologies in the anchoring step, which improves efficiency without loss in effectiveness.

Using BK in ontology matching generates more correct and incorrect candidate mappings. To effectively select the final mappings, we propose two methods: a rule-based one and an ML-based one. For the second method, we designed a set of 27 attributes to enable the use of a ML classification algorithm.

To evaluate our approach, we have conducted extensive experiments with two OAEI tracks in which BK-based ontology matching systems participate: Anatomy and Large-bio. The obtained results show that:

- The BK built with our approach is efficient and effective;
- Our mapping selection methods are effective, and yield almost the same F-measure scores. However, ML-based selection promotes precision, while rule-based selection promotes recall;
- The results of our approach are competitive compared to the state-of-the-art results.

As a future work, we intend to study the possibility of automating, or semi-automating, the ontology preselection process, which should be fast and simple. Indeed, ontology preselection aims at improving the BK selection efficiency. The idea proposed in [46] may be reused for automating ontology preselection. It consists in indexing all ontologies as documents, and to subsequently query the indexed documents using the lexical information of the ontologies to align. However, we believe that new heuristics and methods may be designed to eliminate ineffective ontologies at the outset, especially when dealing with large and multi-domain ontology repositories. For instance, semantically poor ontologies (i.e., those with no concept synonyms or definitions) should be eliminated because they would not be effective to identify new mappings.

In the current approach, the structure of the preselected ontologies is ignored once the BK is built. However, structural information is necessary to derive mappings with various relations. To that end, we plan to enrich the built BK with the semantic relations, such as is-a, part-of, and disjoint, that link the concepts selected from the same ontology, when these relations exist. This will enable to derive mappings with various relations and check their coherence.

12. Acknowledgment

The authors wish to acknowledge the Eiffel Excellence Scholarship program. We thank Ernesto Jimenez Ruiz for sharing the OAEI 2016 LogMapBio ontology selection results. This work was done during a LIRMM-ESI collaboration within the Semantic Indexing of French biomedical Resources (grant ANR-12-JS02-01001) and PraktikPharma

(grant ANR-15-CE23-0028) projects that received funding from the French National Research Agency as well as by the European H2020 Marie Skłodowska-Curie action (agreement No 701771), the University of Montpellier and the CNRS.

13. References

- [1] Zharko Aleksovski, Michel Klein, Warner Ten Kate, and Frank Van Harmelen. Matching unstructured vocabularies using a background ontology. In *15th International Conference on Knowledge Engineering and Knowledge Management, EKAW, Podesbrady, Czech Republic*, pages 182–197, 2006.
- [2] Zharko Aleksovski, Warner Ten Kate, and Frank Van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In *1st International Workshop on Ontology Matching, OM, Athens, Georgia, USA*, pages 13–24, 2006.
- [3] Amina Annane, Zohra Bellahsene, Faical Azouaou, and Clement Jonquet. Selection and combination of heterogeneous mappings to enhance biomedical ontology matching. In *20th International Conference on Knowledge Engineering and Knowledge Management, EKAW, Bologna, Italy*, pages 19–33, 2016.
- [4] Zohra Bellahsene, Vincent Emonet, DuyHoa Ngo, and Konstantin Todorov. YAM++ online: a multi-task platform for ontology and thesaurus matching. In *14th Extended Semantic Web Conference, ESWC, Posters and Demonstrations, Portoroz, Slovenia*, 2017.
- [5] Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:267–270, 2004.
- [6] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [7] Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Roger Granada, Valentina Ivanova, and Ernesto Jiménez-Ruiz. Results of the ontology alignment evaluation initiative 2015. In *10th International Workshop on Ontology Matching, OM, Bethlehem, USA*, pages 60–115, 2015.
- [8] Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In *12th International Semantic Web Conference, ISWC, Sydney, Australia*, pages 294–309, 2013.
- [9] Xi Chen, Weiguo Xia, Ernesto Jiménez-Ruiz, and Valerie Cross. Extending an ontology alignment system with BioPortal: a preliminary analysis. In *13th International Semantic Web Conference, ISWC, Posters and Demonstrations, Riva del Garda, Italy*, pages 313–316, 2014.
- [10] Mathieu d’Aquin and Natalya Fridman Noy. Where to publish and find ontologies? A survey of ontology libraries. *Journal of Web Semantics*, 11:96–111, 2012.
- [11] Hong-Hai Do, Sergey Melnik, and Erhard Rahm. Comparison of schema matching evaluations. In *Web and Database-Related Workshops, Erfurt, Germany*, pages 221–237, 2002.
- [12] Kevin Donnelly. SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in health technology and informatics*, 121:279, 2006.
- [13] Zlatan Dragisic, Valentina Ivanova, Patrick Lambrix, Daniel Faria, Ernesto Jiménez-Ruiz, and Catia Pesquita. User validation in ontology alignment. In *15th International Semantic Web Conference, ISWC, Kobe, Japan*, pages 200–217, 2016.
- [14] Zlatan Dragisic, Valentina Ivanova, Huanyu Li, and Patrick Lambrix. Experiences from the anatomy track in the ontology alignment evaluation initiative. *Journal of Biomedical Semantics*, 8(1):56:1–56:28, 2017.
- [15] Fabien Duchateau and Zohra Bellahsene. YAM: A step forward for generating a dedicated schema matcher. *Transactions on Large-Scale Data and Knowledge-Centered Systems XXV*, 25:150–185, 2016.
- [16] Marc Ehrig, York Sure, and Steffen Staab. Supervised learning of an ontology alignment process. In *3rd Biennial Conference*

- on Professional Knowledge Management/Wissensmanagement, WM, Kaiserslautern, Germany, pages 508–517, 2005.
- [17] Manel Achichi et al. Results of the ontology alignment evaluation initiative 2016. In *11th International Workshop on Ontology Matching, OM, Kobe, Japan*, pages 73–129, 2016.
 - [18] Jérôme Euzenat. Algebras of ontology alignment relations. In *7th International Semantic Web Conference, ISWC, Karlsruhe, Germany*, pages 387–402, 2008.
 - [19] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.
 - [20] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching (second edition)*. Springer, 2013.
 - [21] Daniel Faria, Catarina Martins, Amruta Nanavaty, Daniela Oliveira, Booma Sowkarthiga, Aynaz Taheri, Catia Pesquita, Francisco M Couto, and Isabel F Cruz. AML results for OAEI 2015. In *10th International Workshop on Ontology Matching, OM, Bethlehem, PA, USA*, pages 116–123, 2015.
 - [22] Daniel Faria, Catia Pesquita, Booma S Balasubramani, Catarina Martins, Joao Cardoso, Hugo Curado, Francisco M Couto, and Isabel F Cruz. OAEI 2016 results of AML. In *11th International Workshop on Ontology Matching, OM, Kobe, Japan*, pages 138–145, 2016.
 - [23] Daniel Faria, Catia Pesquita, Emanuel Santos, Isabel F Cruz, and Francisco M Couto. Automatic background knowledge selection for matching biomedical ontologies. *PloS One*, 9(11):e111226, 2014.
 - [24] Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmomari, Isabel F. Cruz, and Francisco M. Couto. The agreementmakerlight ontology matching system. In *On the Move to Meaningful Internet Systems, OTM, Graz, Austria*, pages 527–541, 2013.
 - [25] Anika Groß, Michael Hartung, Toralf Kirsten, and Erhard Rahm. Mapping composition for matching large life science ontologies. In *2nd International Conference on Biomedical Ontology, ICBO, Buffalo, NY, USA*, pages 109–116, 2011.
 - [26] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
 - [27] Michael Hartung, Anika Groß, Toralf Kirsten, and Erhard Rahm. Effective mapping composition for biomedical ontologies. In *9th Extended Semantic Web Conference, ESWC, Heraklion, Crete, Greece*, pages 176–190, 2012.
 - [28] Robert Isele and Christian Bizer. Learning expressive linkage rules using genetic programming. *International Journal on Very Large Data Bases*, 5(11):1638–1649, 2012.
 - [29] Valentina Ivanova and Patrick Lambrix. A unified approach for aligning taxonomies and debugging taxonomies and their alignments. In *10th International Conference the Semantic Web: Semantics and Big Data, ESWC, Montpellier, France*, pages 1–15, 2013.
 - [30] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Alessandro Solimando, and Valerie V. Cross. Logmap family results for OAEI 2015. In *10th International Workshop on Ontology Matching, OM, Bethlehem, PA, USA*, pages 171–175, 2015.
 - [31] Ernesto Jiménez-Ruiz, B Cuenca Grau, and V Cross. LogMap family participation in the OAEI 2016. In *11th International Workshop on Ontology Matching, OM, Kobe, Japan*, pages 185–189, 2016.
 - [32] Ernesto Jiménez-Ruiz, Christian Meilicke, Bernardo Cuenca Grau, and Ian Horrocks. Evaluating mapping repair systems with large biomedical ontologies. In *26th International Workshop on Description Logics, Ulm, Germany*, pages 246–257, 2013.
 - [33] Clement Jonquet, Anne Toulet, and Vincent Emonet. Two years later: the landscape of vocabularies and ontologies in the AgroPortal. In *The International Workshop on sources and data integration in agriculture, food and environment using ontologies, IN-OVIVE, Montpellier, France*, 2017.
 - [34] Patrick Lambrix and Qiang Liu. Using partial reference alignments to align ontologies. In *6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece*, pages 188–202, 2009.
 - [35] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM, 2002.
 - [36] Feiyu Lin and Kurt Sandkuhl. A survey of exploiting WordNet in ontology matching. In *2nd International Conference on Artificial Intelligence in Theory and Practice, IFIP, Milan, Italy*, pages 341–350, 2008.
 - [37] Angela Locoro, Jérôme David, and Jérôme Euzenat. Context-based matching: design of a flexible framework and experiment. *Journal on Data Semantics*, 3(1):25–46, 2014.
 - [38] Mélissa Mary, Lina Soualmia, and Xavier Gansel. Usability and improvement of existing alignments: The LOINC-SNOMED CT case study. In *Knowledge Engineering and Knowledge Management, EKAW, Bologna, Italy*, pages 145–148, 2017.
 - [39] Viviana Mascardi, Angela Locoro, and Paolo Rosso. Automatic ontology matching via upper ontologies: A systematic evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 22(5):609–623, 2010.
 - [40] Mohri Mehryar, Rostamizadeh Afshin, and Talwalkar Ameet. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012.
 - [41] DuyHoa Ngo and Zohra Bellahsene. Overview of YAM++:(not) Yet Another Matcher for ontology alignment task. *Journal of Web Semantics*, 41:30 – 49, 2016.
 - [42] DuyHoa Ngo, Zohra Bellahsene, and Konstantin Todorov. Opening the black box of ontology matching. In *10th Extended Semantic Web Conference, ESWC, Montpellier, France*, pages 16–30, 2013.
 - [43] Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L Rubin, Margaret-Anne Storey, and Christopher G Chute. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37:170–173, 2009.
 - [44] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet: : Similarity - measuring the relatedness of concepts. In *19th National Conference on Artificial Intelligence, San Jose, California, USA*, pages 1024–1025, 2004.
 - [45] Catia Pesquita, Daniel Faria, Emanuel Santos, and Francisco M Couto. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In *8th International Workshop on Ontology Matching, OM, Sydney, Australia*, pages 13–24, 2013.
 - [46] Christoph Quix, Pratanu Roy, and David Kensch. Automatic selection of background knowledge for ontology matching. In *International Workshop on Semantic Web Information Management, SWIM, Athens, Greece*, pages 5:1–5:7, 2011.
 - [47] Shu Rong, Xing Niu, Evan Xiang, Haofen Wang, Qiang Yang, and Yong Yu. A machine learning approach for instance matching based on similarity metrics. In *11th International Semantic Web Conference, ISWC, Boston, USA*, pages 460–475, 2012.
 - [48] Cornelius Rosse and José L.V. Mejino. A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics*, 36(6):478 – 500, 2003.
 - [49] Carlos Rueda, Luis Bermudez, and Janet Fredericks. The mmi ontology registry and repository: A portal for marine metadata interoperability. In *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pages 1–6. IEEE, 2009.
 - [50] Marta Sabou, Mathieu d’Aquino, and Enrico Motta. Exploring the semantic web as background knowledge for ontology matching. *Journal on Data Semantics*, pages 156–190, 2008.
 - [51] Nicholas Sioutos, Sherri de Coronado, Margaret W. Haber, Frank W. Hartel, Wen-Ling Shaiu, and Lawrence W. Wright. NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30 – 43, 2007.
 - [52] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen

- Eilbeck, Amelia Ireland, Christopher J Mungall, et al. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255, 2007.
- [53] Dennis Spohr, Laura Hollink, and Philipp Cimiano. A machine learning approach to multilingual and cross-lingual ontology matching. In *10th International Semantic Web Conference, ISWC, Bonn, Germany*, pages 665–680, 2011.
- [54] Abdel Nasser Tigrine, Zohra Bellahsene, and Konstantin Todorov. Selecting optimal background knowledge sources for the ontology matching task. In *20th International Conference on Knowledge Engineering and Knowledge Management, EKAW, Bologna, Italy*, pages 651–665, 2016.
- [55] Peter Wegner. Interoperability. *ACM Computing Surveys*, 28(1):285–287, 1996.
- [56] Ding Ying and Fensel Dieter. Ontology library systems: The key to successful ontology reuse. In *The first Semantic Web Working Symposium, SWWS, Stanford University, California, USA*, pages 93–112, 2001.