

---

## MAPLE: Metadata-driven orchestration of ontology matching

---

Manuel Fiorelli\* and Armando Stellato

Department of Enterprise Engineering  
Tor Vergata University of Rome  
Roma (RM), Italy  
Email: manuel.fiorelli@uniroma2.it  
Email: stellato@uniroma2.it  
\*Corresponding author

Tiziano Lorenzetti

Lore Star srl  
Roma (RM), Italy  
Email: tiziano.lorenzetti@lorestar.it

**Abstract:** Bringing together disparate datasets on the Semantic Web clearly benefits from ontology matching. Systematic evaluation campaigns have focused on performance, efficiency, scalability and, more recently, human involvement. No less important is the recognition of the differences between datasets, for example in terms of modelling languages, lexicalisation and structure, which enables the selection and configuration of appropriate techniques and support resources. Following the Semantic Web vision of machines dialoguing to solve problems, we propose MAPLE, a framework that semi-automatically orchestrates an alignment plan using metadata about the matched datasets and other available resources. The framework prescribes a metadata profile that combines established vocabularies such as VoID, DCAT, Dublin Core and LIME, making it possible to use metadata accompanying self-describing datasets or published in catalogues. We discuss the integration of the framework into the collaborative knowledge development environment VocBench 3, as well as compatible matching systems.

**Keywords:** ontology matching; metadata; matching scenario; alignment plan; OntoLex-Lemon; LIME; VoID; MAPLE; VocBench 3.

**Reference** to this paper should be made as follows: Fiorelli, M., Stellato, A. Lorenzetti, T. (YYYY) 'MAPLE: Metadata-driven orchestration of ontology matching', *Int. J. Metadata, Semantics and Ontologies*, Vol. X, No. Y, pp.000–000.

**Biographical notes:** Manuel Fiorelli, PhD is a Research Fellow at Tor Vergata University of Rome, researching on knowledge engineering and semantic technologies, and teaching Computer Networks. His current interests include FAIRness, machine-actionability, semantic interoperability, knowledge management and acquisition, and legal knowledge representation. He has authored over 30 publications in workshops, conferences, and journals. In 2022, he joined the program committee of the MTSR conference. Moreover, he is one of the organizers of the LIRAI workshop on legal IR and AI. He is also on the editorial board of the Journal of Web Semantics. Within the W3C Ontology-Lexica Community Group, he contributed to the metadata module LIME. He is a core developer of the platforms VocBench and ShowVoc, both funded by the DIGITAL program. In the field of Legal Informatics, he contributed to the realization of LegalHTML, an HTML domain language for the representation of legal acts on the Web: funded by the Publications Office of the EU, it is being adopted as a new dissemination format for EUR-Lex. He participated in the EU-funded research projects SEMAGROW and KATY.

Armando Stellato, PhD, is Associate Professor at Tor Vergata University of Rome, where he researches and teaches in the fields of Knowledge Engineering and Knowledge Based Systems. He has authored more than 100 publications on conferences and journals in the fields of Semantic Web, NLP and related areas and has been member of the program committees of more than 100 international conferences and workshops. His main interests cover Architecture Design for Knowledge Based Systems, Knowledge Acquisition and Onto-Linguistic interfaces, for which he participated in several EU funded projects and international research initiatives. Professor Stellato is currently leading – under funding from the DIGITAL program – the development of an ecosystem for knowledge acquisition and management, which already produced widely adopted platforms such as VocBench and ShowVoc. He is also active in the field of Legal Informatics, with the realization of LegalHTML, a semantic model for legal acts, funded and under adoption by the

Publications Office of the EU.

Tiziano Lorenzetti, M.S., is technology consultant for Lore Star Srl, working on Semantic Web technologies. He is co-author of 15 scientific publications in the areas of Data and Knowledge Engineering. He is a contributor to relevant open-source Semantic Web platforms for data management and publication, such as Semantic Turkey, CODA, Sheet2RDF, VocBench and ShowVoc. Tiziano Lorenzetti is also IT consultant for the Food and Agriculture Organization (FAO) of the United Nations, where he is involved in the maintenance of the AGRIS bibliographic information system.

## 1 Introduction

The Semantic Web (Berners-Lee, Hendler, & Lassila, 2001) was conceived as an extension of the predominant document web in which the meaning of resources is made explicit, allowing machines – not just humans – to understand them and use them to undertake tasks. The aim was to enable agents – a sort of autonomous programs – to surf the Web and interact with each other, to autonomously accomplish goals exploiting information and services made available on the Web. While the Semantic Web has so far failed to deliver the promised agent-based paradigm, it has succeeded in establishing standards for publishing data on the web (Shadbolt, Berners-Lee, & Hall, 2006). Indeed, the Linked Open Data (LOD) best practices (Berners-Lee, Linked Data, 2006) were precisely aimed at building a Web of Data, enabling the publication, reuse, and integration of data on the Web. In doing so, LOD has reconnected the Semantic Web agenda with the architectural style – based on resources and the links between them – that has allowed the Web to succeed as a distributed hypermedia application at the Internet scale (Fielding & Taylor, 2022).

Ontologies, as “a formal, explicit specification of a shared conceptualisation” (Guarino, Oberle, & Staab, 2009), are a cornerstone of the Semantic Web, as they allow the intended meaning of resources to be expressed unambiguously. However, the construction of a Web of Data inevitably confronted with the development of competing ontologies, or more generally semantic models, for overlapping domains due to differences in the explication or in the conceptualisation. Allowing a certain degree of semantic heterogeneity is indeed positive, as it is associated with autonomy and diversity, while satisfying the complementary requirements of specialisation and experimentation (Wiederhold, 1994). The Semantic Web, which aims to encompass every conceivable domain, must move away from traditional data integration based on an up-front commitment to a mediated schema, since a schema of everything – as required by the Semantic Web – is not achievable and, in any case, very brittle (Madhavan, et al., 2007). In fact, integration in the Semantic Web is an ongoing process, distributed between data publishers (willing to make their datasets maximally reusable) and data consumers (willing to combine disparate datasets), afforded in a pay-as-you-go style (Madhavan, et al., 2007), as the need and benefits of a tighter integration between datasets become clear. This approach to integration is driving a shift from consolidated databases to

dataspaces (Halevy, Franklin, & Maier, 2006). The adoption of Linked Open Data has indeed transformed the Web into a global dataspace (Heath & Bizer, 2011).

Unifying identical identifiers across datasets is the most favourable scenario for data integration, promoted by Linked Open Data, which encourages the reuse of (or at least the assertion of links to) existing ontologies, vocabularies and (ground) resources. When this is not possible for the reasons discussed above, the reconciliation of different ontologies and, more generally, semantic models can be framed as a computational problem thanks to the availability of ontologies and resource descriptions as computational artefacts, available in machine-readable formats.

Ontology matching (Euzenat & Shvaiko, 2013) addresses ontology heterogeneity by constructing alignments between two (or more) ontologies that include correspondences between semantically matching concepts in the matched ontologies. In the following, we adopt a broader interpretation of ontology matching, which includes the discovery and evaluation of alignments between any kind of knowledge resources (e.g., ontologies, thesauri, lexicons) that conform to the core modelling vocabularies (e.g., OWL, RDFS, SKOS, SKOS-XL, OntoLex-Lemon) for the Semantic Web.

Euzenat & Shvaiko (2007) proposed a classification of (schema-based) ontology matching techniques. Among other criteria, it is important to distinguish between *internal* approaches, which are limited to the content of the input ontologies, and *external* approaches, which can make use of other sources of information. The use of background knowledge has been included in a list of challenges for the future growth of the field of ontology matching (Shvaiko & Euzenat, 2013), together with – to mention just a few of them – matcher selection combination and tuning, user involvement, explanation of matching results and alignment management.

The Ontology Alignment Evaluation Initiative<sup>1</sup> (OAEI) has organised annual evaluation campaigns for ontology matching systems for almost twenty years, allowing systematic evaluation of different systems on shared tasks. In addition to focusing on performance (measured in terms of precision, recall or F-measure), the evaluation campaigns have also considered efficiency and scalability – for example, with the introduction of a track on large biomedical ontologies, which pose additional challenges on their own (Faria, et al., 2018) – and, more recently, user involvement – with the introduction of tracks on interactive matching.

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

These campaigns are interested in evaluating different techniques against pre-defined tasks, in a controlled environment, and as such they allow data to be cleaned, uniformed, and generally made “more easily processable” (Fiorelli, Paziienza, & Stellato, 2014; Stellato, 2015). On the contrary, the real-world deployment of ontology matching systems cannot ignore the inherent diversity and unpredictability of matching scenarios, which requires that these systems be flexible enough to adapt to these scenarios with minimal or no human involvement.

An ontology matching system should recognise different matching scenarios that arise from differences between datasets with respect to, for example, modelling languages, lexicalisation, and structure, allowing the selection and configuration of appropriate techniques and support resources.

In this paper, we illustrate the latest evolution of our MAPLE<sup>2</sup> (MAPping Architecture based on Linguistic Evidence) framework that aims at orchestrating an effective matching process through an analysis of the matching scenario at hand, using metadata about the input datasets and other resources that may be suggested as providers of potentially useful external knowledge. With respect to the work that introduced the framework (Fiorelli, Paziienza, & Stellato, 2014), we improved its architecture and provided a use case for it through its integration in the collaborative knowledge development environment VocBench 3<sup>3</sup> (Stellato, et al., 2020), which is the EU-funded successor of VocBench 2 (Stellato, et al., 2015). Under the hood, VocBench 3 is supported by the RDF service platform Semantic Turkey<sup>4</sup> (Paziienza, Scarpato, Stellato, & Turbati, 2012).

This article is an expanded and revised version of the paper (Fiorelli, et al., 2019) we presented at the 13th International Conference on Metadata and Semantics Research (MTSR'19). In addition to a general revision of the content, the main improvements include:

- more detailed and extensive related work,
- expanded and revised description of the architecture, reflecting subsequent developments (which include the possibility to use chains of alignments),
- a description of the “remote alignment services API” for the connection to downstream matching systems,
- extension of the VocBench 3 use case,
- description of some compliant ontology matching systems,
- expansion of the section on discussions.

The article is structured as follows. Section 2 discusses related work. Section 3 presents our framework, while Section 4 presents a use case within a real collaborative editor of ontologies, thesauri, lexicons and datasets in general, complying with relevant Semantic Web and LOD standards. Section 5 describes the integration of some matching systems. Section 6 discusses our work. Finally, Section 7 concludes the article.

## 2 Related work

There are various matching techniques that depend on different assumptions about the structure and content of the input ontologies (or datasets, in our broader interpretation). Since these assumptions may or may not hold for the given matching scenario, it is impossible to identify the best technique in general. For example, exploiting the semantic interpretation of ontologies may not be as effective if the ontologies are axiomatically weak. Similarly, structure-based approaches should perform better on structurally similar ontologies. In fact, techniques are often combined in matching systems. One reason for such combination is to compensate for the respective weakness of different techniques, while benefiting from the strengths of each. In other cases, a technique may need the results of other techniques as an input. This is the case of similarity propagation methods, which, as their name suggests, require an initial similarity vector that is re-evaluated as similarity scores are propagated through an ontology.

However, Li, et al (2009) noted that these ensemble approaches require the ability to select the appropriate techniques, and then fine-tune parameters such as weights and thresholds, to find the “optimal configuration [that] may work for some cases but may not succeed when we change the context”. As the optimisation of a matching strategy is clearly context-dependent, these authors suggest the need for a principled criterion for the selection and fine-tuning of a matching system based on the characteristics of the given matching task.

In the following, we discuss the literature on automatic selection, tuning and combination of matching techniques, covering three main approaches: weighting, rule-based systems, and machine learning.

Li, et al (2009) implemented their ideas in RiMOM a dynamic multi-strategy ontology matching system. Their intuition is to characterise the current matching scenario by means of two metrics computed jointly on the given ontologies:

- *Label similarity*  $F_{LS}(O_1, O_2)$  is the ratio of concepts and properties that share a name;
- *Structure similarity*  $F_{SS}(O_1, O_2)$  is the ratio of non-leaf nodes (both classes and properties) at the same distance from the root of the hierarchy that have the same number of children.

Intuitively, higher label similarity suggests the effectiveness of label comparison in determining the similarity of elements in the ontologies. Similarly, higher structure similarity means that two ontologies are structurally similar, and that structure-based similarity between ontology elements is reliable. Following this intuition,  $F_{LS}(O_1, O_2)$  and  $F_{SS}(O_1, O_2)$  are used to decide the relative weight of lexical and structural approaches,  $w_{name}$  and  $w_{vec}$  respectively, in the underlying similarity measure between ontology elements.  $F_{LS}(O_1, O_2)$  and  $F_{SS}(O_1, O_2)$  also affect the

<sup>2</sup> <https://art.uniroma2.it/maple/>

<sup>3</sup> <https://vocbench.uniroma2.it/>

<sup>4</sup> <https://semanticturkey.uniroma2.it/>

construction of some representations (e.g., the inclusion of structural features in the virtual documents associated with the ontology concepts), and how similarity propagation deals with different edge types.

With some simplification, we can say that the parameter vector  $\vec{\pi}$  is calculated in RiMOM as a function of  $F\_LS(O_1, O_2)$  and  $F\_SS(O_1, O_2)$ :

$$\vec{\pi} = f(F\_LS(O_1, O_2), F\_SS(O_1, O_2))$$

The key insight is that if  $f$  has been previously validated on some test cases, we can reasonably expect the computed parameters to be effective on unseen matching scenarios.

MOMA (Mochol & Jentzsch, 2008) uses a rule-based approach to matcher selection.

MOMA relies on metadata describing candidate matchers and the input ontologies. MOMA defines an ontology of metadata that captures important characteristics of matchers, inspired by the work of Rahm & Bernstein (2001): e.g., whether it is an individual or a combining matcher, the type of input, whether it considers instances, the arity of the output, etc. An online survey was used to obtain a description of existing matchers. MOMA computes metadata about the input ontologies, covering several syntactic features (e.g., the number of specific modelling constructs) and semantic features (e.g., domain, representation language, natural language, level of formality, type of model).

Finally, a set of rules – implemented in SWRL (Horrocks, et al., 2004) – formalises the knowledge about the dependency between characteristics of the matchers and characteristics of the input ontologies. These include obvious compatibility constraints (e.g., only use matchers that can handle the representation languages of the input ontologies), as well as more sophisticated selection rules that somehow define the characteristics of the matchers (e.g., only use constraint-based matchers against sufficiently axiomatised ontologies).

Cruz et al (2012) formulated the configuration of matchers as a classification problem: the class to be predicted is the configuration among a few alternatives. They evaluated different supervised machine learning algorithms, concluding that k-NN gives the best results (due to the lack of training data). The features used to represent the input ontologies as vectors for the learning algorithms are derived from the profiles of the input ontologies. They extended the OntoQA metrics (Tartir & Arpinar, 2007) originally designed for ontology evaluation.

External resources can provide important background knowledge that is not contained in the ontologies being matched. The addition of this missing knowledge can help matching systems to find correspondences across ontologies, through additional synonyms (e.g., "human" in one ontology is a pseudo-synonym of "person" in the other) or conceptual relationships (e.g., humans are mammals).

Faria et al (2014) distinguish external resources as ontologies or thesauri, lexical databases, text corpora and websites. Much of the related work focuses on the use of a single (broad) resource, while in the best cases it describes a specific instance of a generic oracle.

Mascardi et al (2010) identified rules for using upper ontologies in ontology matching through indirect matching experiments using SUMO-OWL, OpenCyc and DOLCE.

BLOOMS (Jain, Hitzler, Sheth, Verma, & Yeh, 2010) uses a bootstrap approach using Wikipedia categories. BLOOMS looks up class names in Wikipedia and associates each of them with a forest (i.e., a collection of trees) obtained from the categories (together with their parents up to a certain level) attached to the returned pages. By comparing the forests associated with different class names in the input ontologies, BLOOMS can find both equivalence and subsumption correspondences.

WikiMatch (Hertling & Paulheim, 2012) also uses Wikipedia, but it only finds (equivalence) correspondences between concepts whose associated search results overlap above a pre-determined threshold. By exploiting inter-lingual links between different editions of Wikipedia, WikiMatch can find cross-lingual correspondences.

Princeton WordNet (Fellbaum, 1998) is a lexico-semantic resource for American English that models word meanings and conceptual relationships. Lin & Sandkuhl (2008) investigated the use of WordNet similarity measures in ontology matching to discover correspondences between concepts with different labels (e.g., "write" and "compose"). S-Match (Giunchiglia, Shvaiko, & Yatskevich, 2004) uses WordNet to implement the novel concept of semantic matching. The idea is to (roughly) formalise labels and concepts from different ontologies as propositional formulas, using WordNet senses as common non-logical symbols, together with the semantic relationships encoded between synsets, thus allowing to find correspondences across ontologies as a reasoning task (specifically, propositional satisfiability – SAT).

Princeton WordNet is specific to English (in fact, its American variant), but it has served as a blueprint for resources in other languages (Bond & Paik, 2012), commonly referred to as wordnets (with the initial in lowercase).

Furthermore, the adoption of the linked data paradigm for the publication of language resources on the web has determined the formation of the Linguistics Linked Open Data (LLOD) cloud (Chiarcos, Nordhoff, & Hellmann, 2012), as a subset of the general LOD cloud. Like its predecessor *lemon* (McCrae, et al., 2012), the OntoLex-Lemon (Cimiano, McCrae, & Buitelaar, 2016) was originally designed to link ontologies to lexicons, but it then became a cornerstone of the LLOD cloud, providing a principled model for representing lexicons (McCrae, Bosque-Gil, Gracia, Buitelaar, & Cimiano, 2017).

We again follow Faria et al (2014) in discussing approaches to discover (or at least automate the selection of) background ontologies to support ontology matchers. Sabou, d'Aquin and Motta (2008) discuss different strategies to discover a correspondence between two candidate classes: they first look up the class names in an ontology search engine, and if no single ontology contains both classes, depending on the strategy, they search – recursively – for correspondences between the neighbours of one of the candidate concepts and the other. Other works find relevant background ontologies

by optimising different metrics that consider the ontologies as a whole:

- similarity between the input ontologies and the background ontology (Quix, Roy, & Kensch, 2011),
- effectiveness of the background ontology (Hartung, Groß, Kirsten, & Rahm, 2012): i.e., the mapping overlap between each input ontology and the background ontology
- mapping gain: i.e., the proportion of new mappings generated using the background ontology (Faria, Pesquita, Santos, Cruz, & Couto, 2014).

### 3 Our framework

**Figure 1** illustrates the architecture of our framework, in terms of interactions and data flows, for orchestrating loosely coupled components for the purpose of configuring a robust ontology matching process. Indeed, this is a framework that needs to be implemented in a hosting platform, which has to connect the different components (e.g., actually implement the data flows and interactions shown in **Figure 1**) and provide a user interface.

The *user* plays an important role in the architecture, as our goal is to implement a semi-automatic process that relieves the user of tedious, repetitive, non-creative work, while at the same time allowing the user to intervene. Far from simply validating the output of a fully automated process, humans can intervene at various times to influence the automated process itself.

The user's first duty is to define (I) a *matching task* consisting of the two datasets ( $D_{\text{left}}$  and  $D_{\text{right}}$ ) to be matched. We adopt a broader definition of the task to cover ontologies, thesauri, and potentially other types of datasets. When a task is defined, the *matching orchestrator* searches (II) for metadata about the input datasets in the *metadata registry*. It also uses the metadata registry to discover potentially useful support resources – currently, wordnets and alignments.

Our *matching orchestrator* looks up metadata about the input datasets in the *metadata registry* and, by analysing the metadata found, becomes aware of the characteristics of the *matching task* (e.g., knowledge/lexicalisation models, overlaps between supported natural languages, potentially useful external resources, etc.) and summarises them in a description of the *alignment scenario* (III), indicating different dimensions and alternative support resources. The user can choose between the alternatives (IV), resulting in a *scenario definition*.

While the choice of a specific *matching approach* is delegated to the downstream matching system, the content of the *scenario definition* is based on general assumptions, such as the use of lexicalisations to seed the matching process, or the search for synonyms or translations within language resources.

The actual execution of the matching process driven by the scenario definition is the responsibility of a downstream alignment service. In fact, another part of the framework is

an *API for remote alignment services* (see Section 3.3), which supports the configuration, setup, and execution of a matching task.

The user can first discover suitable matchers for a given scenario definition (V), which also allows the alignment service to: i) synthesise specific matching strategies (e.g., depending on whether the user has provided a support language resource), ii) provide tailored settings (e.g., hide the parameters related to the combination of different language pairs in the case of monolingual matching). The user can select a matcher and configure its settings, together with system-level settings that are not related to the given scenario definition (VII). The combination of the *scenario definition* with the *system-level and matcher-level settings* results in an *alignment plan* that is submitted to the alignment service for execution (VIII). Alignment tasks are executed asynchronously because they can take time to complete. The user or consuming application must periodically poll the alignment service until the alignment task is successfully completed and an alignment is generated. Users can retrieve the alignment (IX) in the Alignment API format (David, Euzenat, Scharffe, & Trojahn dos Santos, 2011) and validate it (X) if required.

It is noteworthy that steps IV to VI are optional, as the user can opt to use the default option for each step.

The subsequent subsections delve into the detailed description of some architectural features.

#### 3.1 Metadata schema

The *matching orchestrator* prescribes a specific metadata profile, combining popular standards such as DCAT (World Wide Web Consortium (W3C), 2014), VoID (Alexander, Cyganiak, Hausenblas, & Zhao, 2011), OntoLex-Lemon LIME (Fiorelli, Stellato, McCrae, Cimiano, & Pazienza, 2015), Dublin Core (DCMI Usage Board, 2012). The implementation of the actual metadata registry is a responsibility of the hosting environment, such as VocBench 3 as discussed in Section 4.

**Figure 2** shows a description of the EuroVoc<sup>5</sup> thesaurus, which will be used to illustrate the metadata schema. The modelling of a dataset is done as a `void:Dataset` (line 1), and according to VoID we distinguish 4 main categories of metadata:

- *General metadata* (lines 2-3) to determine the usefulness of a dataset in general. In our use case, it primarily aims to aid the development of user interfaces that are more user-friendly. We utilise Dublin Core Metadata Terms to represent the title and description of a dataset. Language tags support multilingualism by permitting different values for the same property in multiple natural languages.
- *Access metadata* (line 4) to locate the actual RDF content of the dataset. Out of the various options available in VoID, we chose to use SPARQL Endpoints.

<sup>5</sup> <http://eurovoc.europa.eu/>

- *Structural metadata* (lines 5-21) to understand how the dataset is structured, and therefore can be queried. First, `void:uriSpace` holds the namespace of the dataset. Then, `dcterms:conformsTo` holds the (knowledge) model, allowing us to distinguish between SKOS thesauri, OWL ontologies, etc. Lines 17-20 illustrate metrics summarising the "size" of the dataset, namely the number of triples, the number of distinct entities described in the datasets, the number of distinct resources appearing as subjects (or objects) of the triples in the dataset. Lines 7-16 illustrate the possibility of describing subsets defined as partitions of the datasets based on a class. In the example, we reuse the same metadata properties to represent the number of instances of different categories of resources (e.g., in the case of a thesaurus, concept schemes, concepts and collections). In fact, line 21 illustrates how a dataset can be related to subsets in general. In the example, these are (sub)datasets introduced to represent other categories of metadata.
- *Linking metadata* (lines 34-39) to understand how a dataset is linked to other datasets. We use the class `void:Linkset`, with properties to represent the source and target dataset, the link predicate and the number of links. Of course, there can be multiple linksets for the same dataset.

Together with an additional section about:

- *Lexicalisation metadata* (lines 23-33) to understand the linguistic information available in a dataset. For this purpose, we use the class `lime:LexicaliationSet` to represent the labels for a given dataset, in a given natural language, using a given lexicalisation model. The latter is analogous to the (knowledge) model introduced earlier and supports the recognition and thus the exploitation of available linguistic information. The description of multilingual resources requires distinct instances of the class `lime:LexicaliationSet` for the each natural language. We then provide metrics such as the total number of lexicalisations, the number of lexicalised resources, the average number of lexicalisations per resource and, finally, the coverage of the lexicalisation set expressed as a percentage (actually a decimal number between 0 and 1) of reference entities that have been lexicalised. In the case of the OntoLex-Lemon model, we also provide a reference to the lexicon and the total number of lexical entries used in the lexicalisations.

To illustrate the description of language resources, we take the example of Open Multilingual Wordnet<sup>6</sup>, shown in **Figure 3**. In fact, we need to start with some background on the representation of language resources using the OntoLex-Lemon model, particularly those that follow the Princeton WordNet blueprint. Their representation is straightforward:

- each synset is mapped to an `ontolex:LexicalConcept` (a kind of `skos:Concept`)
- each word is mapped to an `ontolex:LexicalEntry` (a resource that describes the unit of analysis in a lexicon),

- each sense is mapped to an `ontolex:LexicalSense`.

We thus need three classes at the metadata level to provide a summary description of each group of entities. A `lime:Lexicon` (lines 28-31) describes lexical entries (i.e., words) in each natural language, while an `ontolex:ConceptSet` (lines 26-27) describes the lexical concepts that provide the semantic backbone of the language resource, which can be shared between different natural languages.

Although similar to that of datasets lexicalised using OntoLex-Lemon, the structure of wordnets is defined by the use of lexical concepts and specific properties to associate them with lexical entries through lexical senses. Therefore, we need a dedicated class, called `lime:ConceptualizationSet` (lines 33-41), to describe the binding between a `lime:Lexicon` (describing a set of lexical entries) and an `ontolex:ConceptSet` (describing a set of lexical concepts). The description of a conceptualization set may include metrics like the number of conceptualizations, the number of lexical concepts and entries, as well as the average rates of ambiguity and synonymy (lines 35-39).

### 3.2 Orchestration process

The *matching orchestrator* is indifferent to the technique that is ultimately chosen to compute an alignment; it does, however, make some general assumptions that must hold for such a technique:

- lexicalizations in one or more natural languages provide a fundamental clue for computing an alignment,
- the use of support resources (e.g., wordnets) helps deal with the expressiveness of natural languages (i.e., synonymy) and – in theory – multilingualism (although the latter is still to be implemented at the level of resource discovery).

An additional option was later included to take advantage of existing alignments:

- the combination of alignments is used to identify correspondences between entities.

**Figure 4** illustrates the process on a generic task involving two datasets:  $D_{left}$  and  $D_{right}$ , respectively. Firstly, let us discuss the management of lexicalisation sets, and then we shall move onto chained alignments.

Suppose there are three lexicalisation sets for  $D_{left}$ , one in Italian, one in French and one in British English, while only two lexicalisation sets for  $D_{right}$ , in Italian and British English. The orchestrator then computes the Cartesian product of these two sets, creating 6 possible pairings of lexicalisation sets. Each pairing can be associated with one or more support language resources: currently, only wordnets for monolingual synonym expansion.

The model allows for describing arbitrarily long chains of alignments, from  $D_{left}$  to  $D_{right}$ . Interestingly, each component of the path can be either in its natural "orientation" or in the opposite one: the rationale is that some mapping properties (e.g., `owl:equivalentClass`, `owl:sameAs`, `skos:exactMatch`)

<sup>6</sup> <https://omwn.org/>

are symmetric, so alignments using them are valid in both directions. In addition, some mapping properties have an inverse (e.g. `skos:broadMatch` and `skos:narrowMatch`), making it possible to invert the whole correspondence. However, the current implementation of the orchestrator is limited to paths of length 2, i.e., composition of two alignments.

**Figure 4** on the left shows the *alignment scenario* filled in by the orchestrator with all the possibilities found by the orchestrator in terms of exploitable combinations of lexicalisation sets, language resources and alignment chains. Based on explicit user choices or automatically applied preferences, the scenario is then specialised into a *scenario definition*, as shown in **Figure 4** on the right.

**Figure 5** shows a concrete *alignment scenario* with TESEO<sup>7</sup> and EuroVoc thesauri as the left and right datasets, respectively, and a number of support resources included as *lexicalisation sets*, *chained alignments* and *synonymisers*

In general, we note that the scenario is represented as a JSON object, which - thanks to the adoption of JSON-LD - can be interpreted as RDF without any problem: it is sufficient to map each property of the JSON object to the property of the same name in the adopted metadata vocabularies (see Section 3.1), with the notable exception of `languageTag`, which should be mapped to `lime:language`.

The scenario provides details on the datasets to be matched, via the `leftDataset` and `rightDataset` properties. These properties hold a JSON object that describes each dataset through various properties. Adhering to JSON-LD specifications, the `@id` property supplies a IRI for the dataset (found in the *metadata registry*), which can be used in the rest of the scenario description to reference that dataset. The property `conformsTo` supplies the IRI of the knowledge model for the dataset, which allows to differentiate between an ontology, a thesaurus (as in this case), etc. This distinction is crucial for appropriately interpreting the dataset, as well as establishing alignment goals and supporting the fine-tuning of the matching strategy. For example, aligning two thesauri requires to find correspondences between `skos:Concepts`, using the property `skos:broader` to interpret the concept hierarchy, whereas aligning two OWL ontologies demands correspondences between classes, properties and individuals, using `rdfs:subClassOf` (or `rdfs:subPropertyOf`) for the class hierarchy (or property hierarchy). Finally, the property `sparqlEndpoint` supplies the SPARQL endpoint for query the dataset.

In addition to the datasets to be matched, an alignment scenario may involve other datasets referred to as *support datasets* and thus held by the eponymous property. Their description is more detailed than that of the *left* and *right datasets*, containing additional properties depending on their type (`@type`) as per the metadata schema already described in Section 3.1.

Given the previously stated assumption regarding the role of natural language lexicalisations in comparing different ontologies, the *alignment scenario* provides a list of paired

lexicalisation sets from the two datasets to be aligned in the pairings property. The IRI of each lexicalisation set can serve as an anchor to locate its description in the provided list of *support datasets*. In particular, the `lexicalizationModel` and `languageTag` properties ensure the proper interpretation of the lexicalisation sets with respect to a technical standard for their encoding and the actual natural language they come from. This enables the application of NLP tools, such as lemmatizers, specific to each natural language. Meanwhile, the natural languages associated with the two lexicalisation sets jointly determine whether the scenario at hand is monolingual or multilingual (the latter is under development). Each pair of *lexicalisation sets* may be associated with a `synonymizers` property, which provides zero or more aggregates of an `ontolex:Lexicon` (for the natural language shared between the lexicalisation sets) and a `lime:ConceptualizationSet`, which can be used to compute the synonyms of a word.

The given *pairings* are ranked based on their *score*, which aims to estimate of their utility in computing an alignment. We consider a lexicalisation set, whose left and right sides are denoted as *source* and *target*, respectively. Furthermore, we introduce the (potentially *null*) *lr* variable for a language resource used as synonymizer. The *score* for a pair of lexicalisation sets and a language resource is defined as follows:

$$\begin{aligned} & \text{score}(\text{source}, \text{target}, \text{lr}) \\ &= \text{percentage}(\text{source}) \text{percentage}(\text{target}) (1 \\ & - \alpha e^{-\beta \max(\text{expr}(\text{source}), \text{expr}(\text{target})) (1 + \text{contrib}(\text{lr}, \text{source}, \text{target}))}) \end{aligned}$$

Where:

- *percentage*(*x*) is a function that returns the fraction of the reference dataset covered by the provided lexicalisation set (i.e., the value of the `percentage` property in its description).
- *expr*(*x*) a function estimating the expressiveness lexicalisation set as follows:

$$\text{expr}(x) = \frac{\text{avgNumOfLexicalizations}(x)}{\text{percentage}(x)}$$

The score property of a pairing is computed without a language resource, in which case *contrib*(*lr*, *source*, *target*) is 0. The function has two parameters,  $\alpha$  and  $\beta$ , which have been set to 0.5 and 0.1 respectively.

For each element of the list held by the `synonymizers` property, the score property is computed by taking the element as the value of the *lr* variable. For a non-null language resource, the *contrib*(*lr*) function is so defined:

<sup>7</sup> [http://www.senato.it/3235?testo\\_generico=745](http://www.senato.it/3235?testo_generico=745)

$$\begin{aligned}
& \text{contrib}(lr, source, target) \\
&= \frac{\text{conceptualizations}(\text{conceptualizationSet}(lr))}{\max_{x \in \{source, target\}} \{\text{lexicalizations}(x)\}} \\
&\cdot \text{avgAmbiguity}(\text{conceptualizationSet}(lr)) \\
&\cdot \text{avgSynonymy}(\text{conceptualizationSet}(lr)) \\
&\cdot \frac{\text{lexicalEntries}(\text{conceptualizationSet}(lr))}{\text{lexicalEntries}(\text{lexicon}(lr))}
\end{aligned}$$

Where:

- $\text{conceptualizationSet}(lr)$  returns the conceptualization set associated with a language resource
- $\text{lexicon}(lr)$  is the lexicon set associated with a language resource
- $\text{conceptualizations}$ ,  $\text{lexicalizations}$ ,  $\text{avgAmbiguity}$ ,  $\text{avgSynonymy}$  and  $\text{lexicalEntries}$  are functions that return the value of the metadata properties with the same name in the description of the provided entity.

The highest score associated with the synonymizers is then used as the value the `bestCombinedScore` property of the overall pairing.

Let us analyse the *score* function. We can rewrite it as follows:

$$\text{score}(source, target) = s \cdot t \cdot \left(1 - \alpha e^{-\beta \max\left(\frac{u}{s}, \frac{v}{t}\right)w}\right)$$

where:

- $s = \text{percentage}(source)$  thus  $s \in (0,1]$
- $t = \text{percentage}(target)$  thus  $t \in (0,1]$
- $u = \text{avgNumOfLexicalizations}(source)$  thus  $u > 0$
- $v = \text{avgNumOfLexicalizations}(target)$  thus  $v > 0$
- $w = 1 + \text{contrib}(lr, source, target)$  thus  $w > 1$

First, note that the function is invariant with respect to the swapping of *source* and *target*. Therefore, it is sufficient to study the function with respect to one variable such as  $s$ , without any loss of generality. Building upon this observation, we can demonstrate that the function is increasing in relation to  $s$ .

Considering the presence of the *max* function, we must distinguish between two cases:

1.  $\frac{u}{s} \leq \frac{v}{t}$  : in this case, the *score* function becomes

$$\begin{aligned}
& s \cdot t \cdot \left(1 - \alpha e^{-\beta \max\left(\frac{u}{s}, \frac{v}{t}\right)w}\right) = s \cdot t \cdot \left(1 - \alpha e^{-\beta \frac{v}{t}w}\right) = \\
& sC
\end{aligned}$$

where the second and third factors have been replaced by a positive constant as they do not vary with respect to  $s$

(notice that  $\beta \frac{v}{t}w > 0$ , thus  $e^{-\beta \frac{v}{t}w} < 1$  and finally

$$\alpha e^{-\beta \frac{v}{t}w} < 1 \text{ since } \alpha < 1)$$

2.  $\frac{u}{s} > \frac{v}{t}$  : in this case, the function becomes
- $$s \cdot t \cdot \left(1 - \alpha e^{-\beta \frac{u}{s}w}\right)$$

As  $s$  increases,  $\frac{u}{s}$  decreases; therefore,

- if we are in case 1, we remain in this case, where the score function is clearly increasing with respect to  $s$ ;
- otherwise, we observe that the function can fall into case 1, but first we need to check the behaviour of the function in case 2.

In case 1, the *score* function is clearly increasing in relation to  $s$ ; in case 2, we can check whether  $s \cdot t \cdot \left(1 - \alpha e^{-\beta \frac{u}{s}w}\right)$  is increasing by calculating its partial derivative (with respect to  $s$ ), which is equal to:

$$t - \frac{\alpha t e^{-\frac{\beta u w}{s}}(s + \beta u w)}{s}$$

We first factorise the expression with respect to  $t$ :

$$t \left(1 - \frac{\alpha e^{-\frac{\beta u w}{s}}(s + \beta u w)}{s}\right)$$

Then, we must prove that it is positive (because a differentiable function is increasing if its derivative is positive):

$$t \left(1 - \frac{\alpha e^{-\frac{\beta u w}{s}}(s + \beta u w)}{s}\right) > 0$$

We can then divide both sides for the positive number  $t$ :

$$1 - \frac{\alpha e^{-\frac{\beta u w}{s}}(s + \beta u w)}{s} > 0$$

After some passages, we end up with:

$$\alpha \left(e^{-\frac{\beta u w}{s}} + \frac{\beta u w}{s} e^{-\frac{\beta u w}{s}}\right) < 1$$

Since  $\alpha = 0.5$ , we can rewrite the inequation as follows:

$$e^{-\frac{\beta u w}{s}} + \frac{\beta u w}{s} e^{-\frac{\beta u w}{s}} < 2$$

This inequation is true because:

$$\begin{aligned}
& e^{-\frac{\beta u w}{s}} + \frac{\beta u w}{s} e^{-\frac{\beta u w}{s}} \\
& < \max_{s \in (0,1]} \left\{ e^{-\frac{\beta u w}{s}} \right\} \\
& + \max_{s \in (0,1]} \left\{ \frac{\beta u w}{s} e^{-\frac{\beta u w}{s}} \right\} < 1 + \frac{1}{e} < 2
\end{aligned}$$

(since  $\max_t \{te^{-t}\} = 1/e$ )

To conclude the proof, we must note that there is no discontinuity on the frontier between the two cases, because  $\max\left(\frac{u}{s}, \frac{v}{t}\right) = K$  if  $K = \frac{u}{s} = \frac{v}{t}$ . Otherwise, in the case of a jump discontinuity, for example, we could have had that the function is increasing within each case, but around the



frontier we could have had a lower value on the side of case 2 than that on the side of case 1. In such a scenario, the function would not be monotonic, even though it would be piecewise monotonically increasing.

Let us examine the behaviour of the *score* function with respect to a language resource. It is straightforward to show that the function is again increasing in the contribution of the language resource, which is represented by term  $w$ . As  $w$  increases, the exponent also increases in absolute value. Therefore,  $e^{-\beta \max(\frac{u}{s}, \frac{v}{t})w}$  decreases, and finally  $(1 - \alpha e^{-\beta \max(\frac{u}{s}, \frac{v}{t})w})$  increases.

We conclude the analysis of the scoring function with a qualitative discussion on the factors that occur in the definition of the *contrib*( $lr, source, target$ ) function:

- The first factor relates to likelihood of a sense from either lexicalisation set matching one in the conceptualisation set.
- The second and third factors privilege language resources with high ambiguity and synonymy as more representative of the actual use of the language (i.e., it is better to be know that "dog" is ambiguous than to believe that it has only one sense).
- The fourth factor indicates the proportion of the entries of a lexicon that are actually bound to a concept: between 0 and 1, this value is usually equal to 1.

In a later version of the framework, support was added for calculating correspondences by combining existing alignments. This is achieved through the *alignmentChains* property within the *alignment scenario*, which holds a list of suggested compositions of alignments. These alignments are datasets of type *mdr:Alignment*, which stands for possibly multiple *void:Linksets* between the same pair of datasets with different values for the property *void:linkPredicate*.

In the example, shown in **Figure 5**, the *alignment scenario* involves a match between two thesauri using Italian labels, as it is the only common natural language between them, expressed in SKOS-XL. This monolingual pairing was also associated with two alternative *synonymisers*, obtained using MutiWordNet (Pianta, Bentivogli, & Girardi, 2002) and ItalWordnet (Roventini, et al., 2002) respectively. Both resources were located in the *metadata registry*, which stored information about Open Multilingual WordNet in the given use case. In addition, the scenario proposes to combine two alignments (using a third thesaurus, GEMET<sup>8</sup>, as pivot):

- one from TESEO to GEMET (contained in a third dataset)
- one from EuroVoc to GEMET (contained in the EuroVoc thesaurus)

### 3.3 Remote alignment services API

Our orchestration framework assumes that there exists a downstream alignment service that will execute the

alignment plan. An important requirement is to avoid any commitment to a specific service implementation, and, instead, support the use of diverse technologies. To that end, our framework includes the definition of an *alignment services API*. Following the API-first approach, the framework includes a machine-readable specification of this API in the OpenAPI Format<sup>9</sup>. **Figure 6** shows the specification inside the online Swagger Editor, showing side-by-side, from left to right, the API specification and an automatically generated, interactive documentation.

These API specifications can be versioned, verified, validated, and published. The compliance to the same API specification enables interoperability: requests from clients (e.g., applications willing to do ontology matching) are understood by API implementations (i.e., alignment services), and – vice versa – responses of API implementations are properly interpreted by clients. Compliance to the specifications is supported by code generation tools, which – fed with an API specification – can automatically produce client libraries (to consume the API) and server stubs (providing the skeleton of a compliant API provider). Swagger Codegen<sup>10</sup> and OpenAPI Generator<sup>11</sup> (born as a fork of the former) are two notable examples of such genre of the tools. In fact, the API ecosystem contains a variety of tools for diverse API-related activities, which in most cases can be configured using OpenAPI specifications.

Without entering into the full details of the API, which have been discussed elsewhere for a previous version (Fiorelli & Stellato, 2020), and in any case are available in the specification of the API itself, we give a gist of the API design. The API comprises several endpoints (i.e., URLs) associated with different resources (i.e., a resource-centred design), which can be operated upon uniformly through the HTTP verbs (e.g., GET to retrieve the representation of a resource, POST to create a new resource in a collection, etc.).

Analysing the interactions illustrated in **Figure 1**, we identified the followed resource (collections):

- *root*, which allows to retrieve metadata about the alignment service and to do health checks,
- *matchers*, allowing to retrieve available matchers and their settings. A non-resource (sub)endpoint can be POSTed with a scenario definition, to search for suitable matchers,
- *tasks*, allowing to submit, retrieve e delete tasks.

## 4 Use case: VocBench 3

We integrated our matching orchestration framework into the collaborative knowledge development environment VocBench 3 to validate the soundness of the architecture and to provide a useful solution for end users. As discussed in Section 3, VocBench 3 – as the host environment of the framework – provides:

- an implementation of the *metadata registry*
- a *user interface* that allows:

<sup>8</sup> <https://www.eionet.europa.eu/gemet/en/about/>

<sup>9</sup> <https://www.openapis.org/>

<sup>10</sup> <https://swagger.io/tools/swagger-codegen/>

<sup>11</sup> <https://openapi-generator.tech/>

- select an *alignment service*,
- enter *system-level settings*,
- define a *matching task*,
- *profile* a task,
- refine a matching scenario into a *scenario definition* which, together with optional *matcher settings*, is an *alignment plan* for the alignment service,
- *validate* the generated alignment,
- message delivery according to the data flows shown in **Figure 1**.

**Figure 7** shows the user interface for creating a task to align the EuroVoc and TESEO thesauri. After selecting the right dataset (i.e., as in this scenario, the left dataset corresponds to the current project), and making sure that the metadata is available and up to date (i.e., clicking on the histogram icon next to each dataset name), the user can profile the matching problem, to obtain a scenario definition. At the top, there are the descriptions of the input datasets, followed by a sorted list of paired lexicalisation sets, each associated with zero or more language resources. The default strategy for obtaining an actual scenario definition is to select the first pairing and no language resource. However, the user can easily change these choices. Further customisation is possible by searching for a suitable matcher (without relying on the implicit choice made by the alignment service), possibly acting on its specific settings.

The VocBench 3 metadata registry supports several strategies for capturing dataset metadata:

- manual addition of a (remote) dataset description,
- discovery of a (remote) dataset description using the VoID backlink mechanism,
- limited profiling of (remote) dataset SPARQL endpoints,
- harvesting of (automatically generated) metadata about local projects.

The VocBench 3 use case is concerned with the alignment of locally hosted resources – mainly for performance reasons – for which the fourth strategy is particularly convenient.

The user who has submitted a task is not blocked while waiting for it to be completed – which can take a lot of time for large inputs – but instead can do other work and even log out from the system.

When an alignment task has been successfully completed, the generated alignment can be opened within a validation panel (see **Figure 8**). The generated correspondences are listed one by one, showing the related entities and their score (either as a progress bar or as a number). To check the correctness of a correspondence, the user can click on one of the related resources to open its description in a modal resource view. Indeed, the user can reject or accept individual correspondences: in the latter case, it is possible to choose a more specific mapping relation depending on the nature of the related resource (e.g., equivalence between SKOS concepts can be detailed as `skos:exactMatch` or `skos:closeMatch`). Another option is bulk validation by automatically applying a validation criterion to each correspondence: e.g., accept all correspondences whose score exceeds a user-defined threshold.

The alignment can be exported in the Alignment API format (David, Euzenat, Scharffe, & Trojahn dos Santos, 2011), with extensions to keep track of the validation status and the refinement of the mapping relation. In fact, a (previously exported) alignment file can be loaded into the same validation panel, e.g., to resume the validation process from the point where it was previously interrupted.

Another possibility is to apply the alignment to the left dataset:

- the triples for accepted correspondences are added to the dataset,
- optionally, the triples for rejected correspondences can be deleted (if the incorrect correspondences were already asserted in the dataset).

## 5 Compliant alignment services

We have collaborated with the authors of a number of matching systems to integrate them with our framework. In fact, the integration of these system only required the development of a server that implements our alignment services API (see Section 3.3). Our goal was facilitated by the fact that these systems were designed to be used as libraries or were already designed with a REST API. In the following sections, we briefly mention the currently compatible systems.

### 5.1 Genoma

Genoma (Enea, Pazienza, & Turbati, 2015) is an ontology matching environment that provides the user with a powerful tool for designing and testing ontology matching architectures. Genoma allows defining complex matching architectures as dataflows. A similarity matrix is first extracted by a *similarity matrix extractor*, then different similarity matrices (e.g., based on element-level features or structural features) can be combined into a single matrix at a *junction point*. An *alignment extractor* produces an alignment by filtering a similarity matrix. Finally, a *graph extractor* produces the final alignment graph.

### 5.2 Matcha

Matcha (Faria, Contreiras Silva, Cotovio, Eugénio, & Pesquita, 2022) is a new ontology matching system that is being developed as a successor to the successful AgreementMakerLight (AML) (Faria, et al., 2013). The motivation for developing a new system lies in the need for a completely new architecture to better address new requirements that have become particularly relevant for biomedical ontology use cases: holistic matching (i.e., aligning multiple ontologies), complex ontology matching (i.e., relating non-atomic concepts between ontologies), and, finally, the adoption of machine learning approaches.

### 5.3 Naisc

Naisc (McCrae & Buitelaar, 2018) is an automated linking tool developed at the Insight Centre for Data Analytics. 'Naisc' means 'links' in Irish and is pronounced 'nashk'. Naisc's modularity allows for extensions by implementing different component types, such as lenses, feature extractors, scorers, matchers and constraints. Supporting blocking for scalability, Naisc combines textual and graph features to determine similarity scores, while the final alignment is derived under customizable constraints (e.g., bijectivity).

### 5.4 ST Remote Service Compendium

The ST Remote Service Compendium<sup>12</sup> is a collection of utility services for Semantic Turkey. In fact, these services share the fact that they can take a lot of time for large inputs; therefore, they are best modelled as long-running, asynchronous tasks that are handed off to a dedicated task scheduling application. Furthermore, offloading these tasks to a separate application satisfies the need for replacing the "predefined" implementation with an alternative one, e.g., using different algorithms, making different assumptions, committing to a different compliance/performance trade-off.

One such utility is the so-called *alignment bootstrap*, which generates an alignment between datasets bridged by alignments to a third one.

## 6 Discussion

The integration of our framework into VocBench 3, discussed in Section 4, shows the soundness of our approach and its usefulness in real applications. Moreover, this integration can be considered as another contribution in its own right, intended as an end-user application, freely available at no cost, and certainly not limited to a proof-of-concept experiment.

Let us discuss the main tenets of the proposed architecture, which has been described in Section 3.

It clearly decouples the *matching orchestrator* from the generation of metadata as the former only consults the *metadata registry* for dataset descriptions, regardless of how they were created. This separation of concerns allows for the adoption of different strategies: for example, in the VocBench 3 use case, we accommodate both locally managed datasets and others on the web, possibly accompanied by descriptive metadata. However, the framework provides reusable components (e.g., a dataset profiler) that can support the implementation of different metadata harvesting strategies.

Our framework assumes that in most cases ontology matching is ultimately based on the similarity of the linguistic information associated with each ontology, primarily the labels. Accordingly, the *alignment scenario* returned by our framework contains pairs of *lexicalisation sets* that can be used as a potential basis for comparison. The comparison can

certainly benefit from the use of language resources such as dictionaries, wordnets or domain terminologies, which can help to handle synonymy, ambiguity and even suggest potential semantic relationships in the case of lexico-semantic resources (e.g., wordnets). Most related work uses pre-defined resources that come with the ontology matching system. In fact, some systems download them on the fly from hardcoded locations, allowing for a smaller download size for the matching systems, and potentially a smaller footprint if only resources used are downloaded. Customisation of the language resources is rarely considered a priority and, if possible, requires complex configuration or low-level tinkering with the system. Conversely, in our framework, appropriate language resources are suggested (e.g., as synonymizers) next to each *pair of lexicalisation sets*, after they have been identified by querying the *metadata registry*. Accordingly, our framework does not commit to or suggest a specific language resource, with the advantage of being domain and language agnostic. In contrast to other systems that rely on the native format of each language resource, our framework relies on OntoLex-Lemon as a unified model, a de facto standard that is actually used for publishing language resources on the Web, with the obvious advantage of making them potentially usable in the context of our framework. In fact, our approach to language resources is closer in spirit and practical outcome to related work focused on background ontology retrieval, discussed later in this section.

Language resources are treated as *support datasets*, which are (potentially) separate from the ontologies to be matched. In fact, the *lexicalisation sets* mentioned above are also represented as *support datasets* that can be contained in any input ontology, but more interestingly, they can be accessed as a third-party resource (discovered in the *metadata registry*). This distinguishing feature of our framework with respect to most matching systems is in line with the more diverse publication strategies associated with OntoLex-Lemon. For example, it is quite common for an OntoLex-Lemon lexicon to be published separately from a (pre-existing) dataset, in order to provide a richer linguistic characterisation of the dataset (than what is conveyed, for example, by the RDFS labels within it), or to support another natural language.

In its later evolution, our framework also supports matching by composing already existing alignments between pairs of datasets, which is another example of using background knowledge. These alignments are, again, found in the metadata registry, without assuming any containment relationship between the alignment (intended as a collection of triples connecting related entities) and the datasets they target.

Our framework relies on the hosting environment implementing a *metadata registry* to efficiently identify relevant datasets by querying structured metadata. Obviously, the registry needs to be pre-fed with information about potentially useful datasets. However, it is possible to extend the registry incrementally (always before querying),

<sup>12</sup> <https://bitbucket.org/art-uniroma2/st-rsc/>

as shown by the use of VoID backlinks to dataset descriptors, in the VocBench 3 use case. This use case also shows an advantage of our reliance on a combination of existing metadata vocabularies, making it possible to reuse, almost unchanged, metadata published alongside a dataset or provided by dataset catalogues.

The use of explicit metadata describing the ontology to be matched and, unlike us, the available matchers, is something that MOMA shares with our framework. For example, on the one hand, MOMA recognises the diversity between different resources, such as OWL ontologies and SKOS thesauri, just like our framework; however, since it is mainly concerned with constraint checking, MOMA only checks whether a matcher can handle a certain type of resource, whereas our goal is to support the adaptation of the same matcher to different resource types.

The description of the *alignment scenario* also considers metrics about the ontologies to be matched and other support datasets. These metrics mostly describe intrinsic characteristics of these datasets and can therefore be computed for each dataset separately, once and for all, independent of other datasets. Conversely, RiMOM computes metrics about the matching task jointly over the two ontologies to be matched (e.g., similarity of the labels between the two ontologies).

Regarding the coverage of the metadata model, MOMA covers probably the most of syntactic (i.e., modelling constructs) and semantic features (e.g., subject domain, level of formality, natural language). On the other hand, OntoQA excels in the level of detail concerning the ontology structure. However, it only considers lexicalisation through a single metric called "readability", which is linked to the existence of `rdfs:labels` and `rdfs:comments`. This implies that OntoQA ignores the other lexicalisation models besides RDFS and even the actual natural language used.

As previously stated, the selection of language resources in our framework is related to the selection of background ontologies. Previous studies have framed this task as an optimisation problem with respect to several metrics (similarity, effectiveness, mapping gain) formulated with respect to one or both of the ontologies to be matched.

Our approach to language resource selection uses a different two-step process. First, the available language resources are filtered to exclude irrelevant ones by looking at the metadata (primarily the natural language) to determine their suitability for the given matching task. For example, an Italian wordnet is suitable for a scenario of matching two datasets, both lexicalised in Italian, whereas a multilingual wordnet would be selected in the case of cross-lingual matching. In a subsequent step, the candidate language resources are ranked according to their intrinsic characteristics (i.e., independent of the matching scenario), giving preference to larger resources.

In contrast to the related research on background ontology selection that accounts for actual overlap in data, our two-step process operates exclusively at the metadata level. Our methodology is most suitable for homogeneous resources such as general vocabularies, as their usefulness typically

increases with their size. This assumption does not apply when candidate resources contain domain-specific resources. For instance, there is no reason to believe that a larger medical terminology would be better suited to the alignment of two business thesauri than a smaller resource for that specific domain. An interesting subject for future investigation is whether this problem can be solved with additional metadata, such as a resource domain, or by utilising lexical overlap metrics. These metrics are evidently context-dependent, as they are computed on a language resource against the specific datasets to be matched. Therefore, they are unlikely to be part of a language resource description in the *metadata registry*, let alone its online description. In fact, these metrics are rather concerned with the alignment scenario.

Defining an API to provide a *common interface with remote alignment services* is an important aspect of our framework. This could be associated with the REST API provided by the Alignment Server, included with the Alignment API. The Alignment Server oversees an ontology network, and, as a result, handles the computation, validation, storage, and retrieval of alignments. In contrast, our API solely deals with transferring the execution of an alignment plan to a remote alignment service, while other needs related to the management of alignments are met by incorporating the framework into a host environment, such as the one outlined in Section 4. The integrated system described in this section is aligned with the capability of GOMMA, which is a "generic infrastructure for managing and analysing life science ontologies and their evolution" (Kirsten, Gross, Hartung, & Rahm, 2011). Our system meets capabilities like storage of (versions of) ontologies and alignments, as well as the invocation of alignment services. Support for diffing is provided for SKOS and for generic RDF datasets by an external service that returns the changes found in two versions of a dataset. The service does not rely on any a priori differential storage or history, and instead analyses any two datasets provided as input, with one marked as the source version and the other as the evolved version. The change report varies depending on the semantic model of the dataset: instead of just reporting changed triples, it tries to abstract several changed triples into macroscopic operations, e.g. in SKOS two triples (one added and one removed) sharing the same subject, the predicate `skos:prefLabel` and having different literals as subject, with the same language tag, would be interpreted as a single change, reported as a change of the resource's preferred label. Triple changes are therefore first collected in a pool and then consumed by these high-level descriptors to produce human-readable reports. The remaining changes are reported as atomic triple changes.

Alignments can also be maintained through the system. One of the Integrity Constraint Validators (i.e. a set of modules checking the integrity of the content with respect to the semantics of the various RDF languages and of the allowed configurations of the data) reports broken alignments, i.e. alignments to external resources that are either non-existing or deprecated. The check for existence is broad, as the target resource can be hosted privately into

VocBench itself or even hosted on the web: VB will in any case treat any aligned dataset as a linked open dataset and thus looked upon the Web. Deprecated resources are not wrong per se, but raise an alert so that a better resource replacing them could be looked for in the target dataset.

The need to interact with diverse systems through a unified interface is shared by infrastructures such as SEALS (Semantic Evaluation At Large Scale) (Gutiérrez, García-Castro, & Gómez-Pérez, 2010) and HOBBIT (Holistic Benchmarking of Big Linked Data) (Röder, Kuchelev, & Ngonga Ngomo, 2019), which support the systematic evaluation of semantic technologies by running them against shared test cases. In accordance with their objectives, these infrastructures address requirements such as packaging, deployment, and resource provision for running the system under test. These requirements are beyond the scope of our API, while the infrastructures just mentioned disregard a key requirement of our framework, which is to support, and in a sense even guide, the execution of an alignment system by providing an alignment plan.

Hertling, Portisch & Paulheim (2019) introduced MELT (Matching Evaluation Toolkit), a software toolkit that simplifies the development, configuration, evaluation, and packaging of ontology matchers. Developers can use it to evaluate their own systems and fine-tune parameters by grid search. It also makes it easy to package the systems for use in the SEALS and HOBBIT infrastructures. MELT defines numerous Java interfaces for the development of matchers. These interfaces are dedicated to significant concepts such as matchers that compute correspondences and filters that further process correspondences that have already been computed. Furthermore, MELT provides various pre-defined implementations of each interface. In this respect, MELT targets a similar use case as GENOMA. MELT additionally provides interfaces for connecting labels to concepts specified by external resources that could have varying degrees of support for dealing with synonyms and hypernyms. This feature streamlines the creation of matchers that accept diverse external resources, but resource selection is not addressed. MELT offers YAAA (Yet Another Alignment API) as a replacement for the popular Alignment API, bringing improvements, such as improved Maven support and customisable indexes to accelerate arbitrary queries over alignment. Apart from developing matchers as software components to be used locally, MELT enables the development of matching systems exposed as web services by defining a standardised API. The latter defines only an operation for matching to take place. The primary arguments, namely the source and target ontology and a reference alignment, can be either provided as a URL or brought into the request body. On the other hand, our API depends on the use of SPARQL endpoints for the ontologies to be matched and other support datasets. Moreover, we have implemented a ticket-based approach to allow for non-blocking computation of alignments, while the MELT API is blocking. The MELT API allows for additional parameters to be included in the request to the match operation. However, the API lacks our support for querying the matchers for

supported configuration options. Nonetheless, it defines a number of predefined parameters that could be implemented by different systems. These parameters include some that specify the natural language of the input ontologies, such as *sourceLanguage* and *targetLanguage*, clearly assuming that there is one primary natural language. In contrast, our API supports *multilingual lexicalisations* in an *alignment scenario*.

While our API appears RESTful on the surface, it does not fully comply with this architectural style. It lacks a hypermedia format and, more specifically, it fails to satisfy the HATEOAS (Hypermedia as the Engine of Application State) constraint (Fielding R. T., 2008). Many web APIs share this limitation, which led to the creation of the dedicated term for such design: pragmatic REST. Although lacking links in messages exchanged at runtime, we utilized a feature of the OpenAPI format to specify some links in the API's specification. The intention is to explain how information contained in the response to a request can be used as an input for a different request, thereby exposing the protocols underlying the use of the API.

Lastly, we show the significance of our work by aligning its capabilities with some of the eight challenges for the sustainable growth of the ontology matching field proposed by Shvaiko and Euzenat (2013).

- *Matcher selection combination and tuning.* The downstream ontology matchers receive a description of the alignment scenario with the request for its configuration parameters, allowing the systems to dynamically generate a suitable configuration. Moreover, the alignment service can adapt and fine-tune itself to fit the characteristics of the given alignment scenario when the alignment plan is sent to the matcher for the actual run. The information contained therein is helpful for achieving this.
- *User involvement.* We have adopted a semi-automated process, with optional human involvement in the definition of an alignment plan and subsequent validation of the generated alignment.
- *Explanation of matching results.* Our approach rather emphasises visibility into the process of setting up and configuring the alignment service for a given task.
- *Alignment management: infrastructure and support.* Our REST API and, even more, the overall use case discussed in Section 4 provide an infrastructure to manage alignment computation, validation, and storage.

## 7 Conclusions

Ontology matching tasks differ greatly in the range of variability along dimensions such as modelling language, lexicalisation and structural patterns. It is therefore necessary to recognise the different matching scenarios that arise from these differences in order to select and setup an appropriate matching process. Ideally, this should be achieved with limited human supervision, supported as much as possible by automation.

We have designed our metadata-driven framework MAPLE for matching orchestration with these needs in mind and broadening its scope to encompass a variety of matching scenarios related to dataset types other than ontologies. The orchestrator relies on a metadata registry containing metadata about the input datasets and other available resources (mainly wordnets) that can be selected as candidate support resources in the matching process. The joint analysis of this metadata enables the recognition of the current alignment scenario. This is then refined into a scenario definition which, together with optional system and matcher settings, forms part of the alignment plan that is delivered to a downstream alignment service that will execute the matching process. The plan will guide the service in the correct interpretation of the input datasets in terms of their semantic and lexical model. In addition to supporting the use of information contained within the input datasets, the plan may include support resources such as language resources or existing alignments that bridge the input datasets. In fact, lexicalisations can – in principle – be provided by external resources. Unlike most systems that rely on hard-coded resources, our framework fully adheres to the Semantic Web vision, where useful resources are intelligently discovered and exploited with a strong reliance on explicit metadata.

This intelligent orchestration of a matching process should extend to the software agents, not just limited to the data. To this end, our framework allows the participation of not previously anticipated systems through the commitment to an explicitly specified API. Thanks to collaborations with different research groups, we succeeded in having implementations of our API for some systems, including top-notch ones.

By its very nature, our framework needs to be integrated into a hosting environment that provides a metadata registry, a user interface and all the necessary wiring between the loosely coupled parts of the architecture. We have also addressed this need by embedding the framework within the collaborative knowledge development environment VocBench 3. The result is a highly configurable, yet highly usable, ontology matching experience that can – in fact – deliver state of the art performance as it can benefit from a growing library of compatible matching systems.

## Acknowledgments

This work has been carried on in the scope of the ISA<sup>2</sup> Programme (<https://ec.europa.eu/isa2/>) funded projects VocBench and PMKI. Recent improvements have also been supported by the KATY project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017453. Furthermore, we acknowledge the support from Project ECS 0000024 Rome Technopole, – CUP B83C22002820006, NRP Mission 4 Component 2 Investment 1.5, Funded by the European Union – NextGenerationEU.

## References

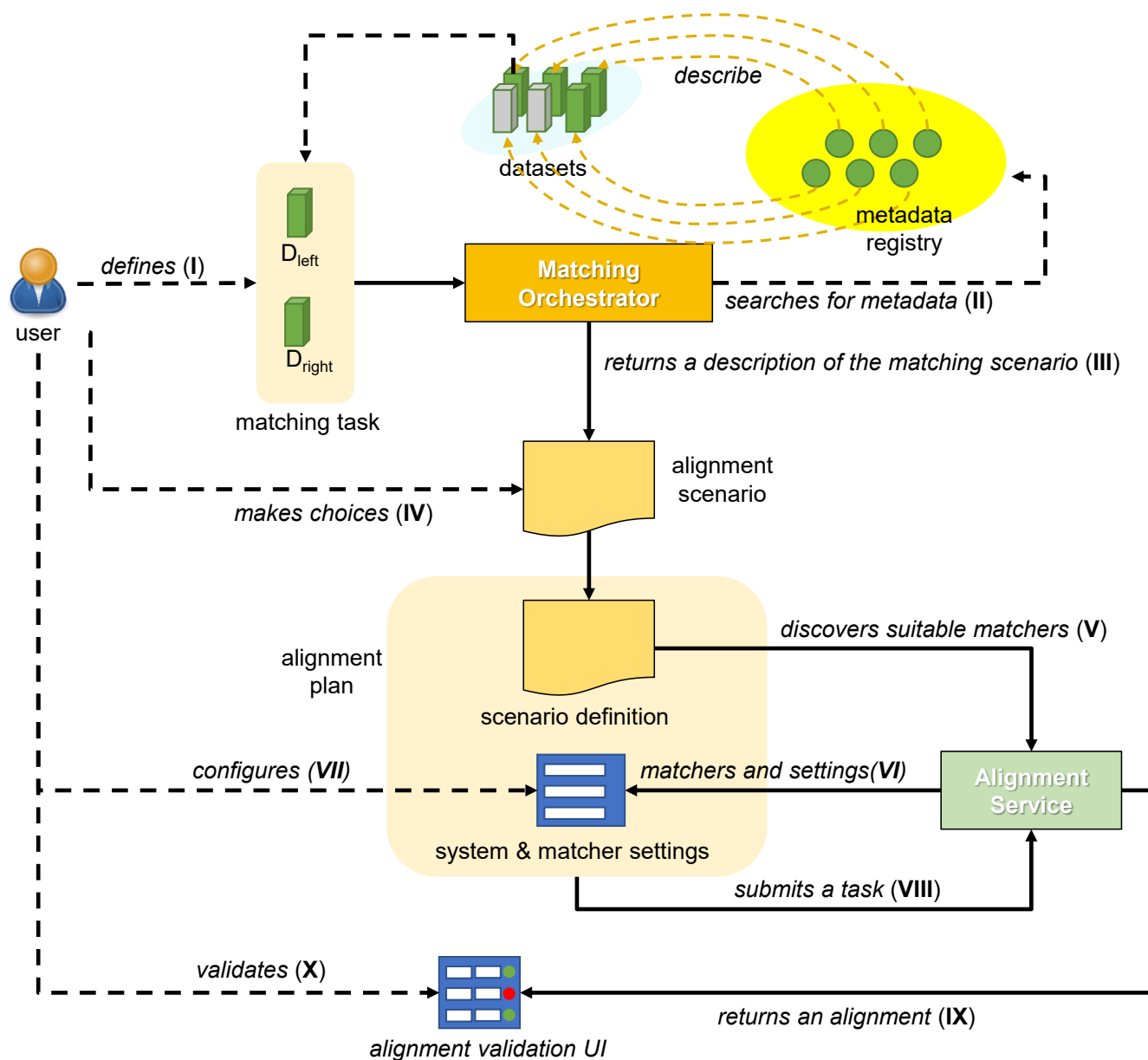
- Alexander, K., Cyganiak, R., Hausenblas, M., & Zhao, J. (2011, March 3). *Describing Linked Datasets with the VoID Vocabulary (W3C Interest Group Note)*. Retrieved May 16, 2012, from World Wide Web Consortium (W3C): <http://www.w3.org/TR/void/>
- Berners-Lee, T. (2006). *Linked Data*. Retrieved November 9, 2017, from Design Issues: <https://www.w3.org/DesignIssues/LinkedData.html>
- Berners-Lee, T., Hendler, J. A., & Lassila, O. (2001). The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5), 34-43. doi:10.1038/scientificamerican0501-34
- Bond, F., & Paik, K. (2012). A survey of wordnets and their licenses. In 2012 (Ed.), *Proceedings of the 6th Global WordNet Conference (GWC 2012)*. Matsue, Japan, January, 9-13, 2012, (pp. 64-71).
- Chiarcos, C., Nordhoff, S., & Hellmann, S. (A cura di). (2012). *Linked Data in Linguistics*. Springer.
- Cimiano, P., McCrae, J. P., & Buitelaar, P. (2016). *Lexicon Model for Ontologies: Community Report, 10 May 2016*. Community Report, W3C. Retrieved from <https://www.w3.org/2016/05/ontolex/>
- Cruz, I. F., Fabiani, A., Caimi, F., Stroe, C., & Palmonari, M. (2012). Automatic Configuration Selection Using Ontology Matching Task Profiling. In E. Simperl, P. Cimiano, A. Polleres, O. Corcho, & V. Presutti (Eds.), *The Semantic Web: Research and Applications. ESWC 2012 (Lecture Notes in Computer Science)* (Vol. 7295, pp. 179-194). Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-30284-8\_19
- David, J., Euzenat, J., Scharffe, F., & Trojahn dos Santos, C. (2011). The Alignment API 4.0. *Semantic Web Journal*, 2(1), 3-10.
- DCMI Usage Board. (2012, June 14). *DCMI Metadata Terms*. Tratto il giorno March 4, 2013 da Dublin Core Metadata Initiative (DCMI): <http://dublincore.org/documents/dcmi-terms/>
- Enea, R., Pazienza, M. T., & Turbati, A. (2015). GENOMA: GENeric Ontology Matching Architecture. In M. Gavanelli, E. Lamma, & F. Riguzzi (A cura di), *Lecture Notes in Computer Science* (Vol. 9336, p. 303-315). Springer International Publishing. doi:10.1007/978-3-319-24309-2\_23
- Euzenat, J., & Shvaiko, P. (2007). Classifications of ontology matching techniques. In *Ontology Matching* (pp. 61-72). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-49612-0\_4
- Euzenat, J., & Shvaiko, P. (2013). *Ontology Matching* (2 ed.). Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-642-38721-0
- Faria, D., Contreiras Silva, M., Cotovio, P., Eugénio, P., & Pesquita, C. (2022). Matcha and Matcha-DL results for OAEI 2022. In P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh, & C. Trojahn (Ed.), *Proceedings of the 17th International Workshop on Ontology Matching (OM 2022), Hangzhou, China, held as a virtual conference, October 23, 2022* (pp. 197-201). CEUR-WS.org. Retrieved from [https://ceur-ws.org/Vol-3324/oei22\\_paper11.pdf](https://ceur-ws.org/Vol-3324/oei22_paper11.pdf)
- Faria, D., Pesquita, C., Mott, I., Martins, C., Couto, F. M., & Cruz, I. F. (2018). Tackling the challenges of matching biomedical ontologies. *Journal of Biomedical Semantics*, 9(4). doi:10.1186/s13326-017-0170-9
- Faria, D., Pesquita, C., Santos, E., Cruz, I. F., & Couto, F. M. (2014). Automatic Background Knowledge Selection for Matching Biomedical Ontologies. *PLOS ONE*, 9(11), 1-9. doi:10.1371/journal.pone.0111226

- Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., & Couto, F. M. (2013). The AgreementMakerLight Ontology Matching System. In R. Meersman, H. Panetto, T. Dillon, J. Eder, Z. Bellahsene, N. Ritter, . . . D. Dou (Eds.), *OTM 2013: On the Move to Meaningful Internet Systems: OTM 2013 Conferences (Lecture Notes in Computer Science)* (pp. 527–541). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-41030-7\_38
- Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: WordNet Pointers, MIT Press.
- Fielding, R. T. (2008, October 20). *REST APIs must be hypertext-driven*. Retrieved from Untangled: <https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>
- Fielding, R. T., & Taylor, R. N. (2022). Principled design of the modern Web architecture. *ACM Transactions on Internet Technology*, 2(2), 115–150. doi:10.1145/514183.514185
- Fiorelli, M., & Stellato, A. (2020). A Lime-Flavored REST API for Alignment Services. *European Language Resources Association* (pp. 52–60). European Language Resources Association. Retrieved from <https://aclanthology.org/2020.ldl-1.8>
- Fiorelli, M., Pazienza, M. T., & Stellato, A. (2014, May). A Metadata Driven Platform for Semi-automatic Configuration of Ontology Mediators. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, . . . S. Piperidis (Ed.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA).
- Fiorelli, M., Stellato, A., Lorenzetti, T., Schmitz, P., Francesconi, E., Hajlaoui, N., & Batouche, B. (2019). Metadata-driven Semantic Coordination. In E. Garoufallou, F. Fallucchi, & E. William De Luca (Eds.), *Metadata and Semantic Research (Communications in Computer and Information Science)* (Vol. 1057). Springer, Cham. doi:10.1007/978-3-030-36599-8\_2
- Fiorelli, M., Stellato, A., McCrae, J. P., Cimiano, P., & Pazienza, M. T. (2015). LIME: the Metadata Module for OntoLex. In F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, & A. Zimmermann (Eds.), *The Semantic Web. Latest Advances and New Domains (Lecture Notes in Computer Science)* (Vol. 9088, pp. 321–336). Springer International Publishing. doi:10.1007/978-3-319-18818-8\_20
- Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2004). S-Match: an Algorithm and an Implementation of Semantic Matching. In T. S. Science, C. J. Bussler, J. Davies, D. Fensel, & R. Studer (Eds.), *The Semantic Web: Research and Applications* (Vol. 3053, pp. 61–75). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-25956-5\_5
- Guarino, N., Oberle, D., & Staab, S. (2009). What Is an Ontology? In *Handbook on Ontologies* (pp. 1–17). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-92673-3\_0
- Gutiérrez, M. E., García-Castro, R., & Gómez-Pérez, A. I. (2010). Executing evaluations over semantic technologies using the SEALS Platform. *Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010)*. Shanghai, China: CEUR-WS.org. Retrieved from <http://ceur-ws.org/Vol-666/paper11.pdf>
- Halevy, A., Franklin, M., & Maier, D. (2006). Principles of Dataspace Systems. *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (pp. 1–9).
- Hartung, M., Groß, A., Kirsten, T., & Rahm, E. (2012). Effective mapping composition for biomedical ontologies. *Workshop on Semantic Interoperability in Medical Informatics (SIMI)* (pp. 176–190). Springer, Berlin, Heidelberg. doi:10.1007/978-3-662-46641-4\_13
- Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1), 1–136. doi:10.2200/S00334ED1V01Y201102WBE001
- Hertling, S., & Paulheim, H. (2012). WikiMatch: Using Wikipedia for Ontology Matching. *Proceedings of the 7th International Conference on Ontology Matching - Volume 946* (pp. 37–48). CEUR-WS.org. Retrieved from [http://ceur-ws.org/Vol-946/om2012\\_Tpaper4.pdf](http://ceur-ws.org/Vol-946/om2012_Tpaper4.pdf)
- Hertling, S., Portisch, J., & Paulheim, H. (2019). MELT - Matching Evaluation Toolkit. In M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, & Y. Sure-Vetter (Eds.), *SEMANTICS 2019: Semantic Systems. The Power of AI and Knowledge Graphs (LNISA)* (Vol. 11702, pp. 231–245). Cham: Springer. doi:10.1007/978-3-030-33220-4\_17
- Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004, May 21). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Retrieved November 29, 2023, from <https://www.w3.org/submissions/SWRL/>
- Jain, P., Hitzler, P., Sheth, A. P., Verma, K., & Yeh, P. Z. (2010). Ontology Alignment for Linked Open Data. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhan, J. Z. Pan, . . . B. Glimm (Eds.), *The Semantic Web – ISWC 2010. ISWC 2010 (Lecture Notes in Computer Science)* (Vol. 6496, pp. 402–417). Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-17746-0\_26
- Kirsten, T., Gross, A., Hartung, M., & Rahm, E. (2011). GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. *Journal of Biomedical Semantics*. doi:10.1186/2041-1480-2-6
- Li, J., Tang, J., Li, Y., & Luo, Q. (2009). RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *IEEE Transactions on Knowledge and Data Engineering*, 21(8), 1218–1232. doi:10.1109/TKDE.2008.202
- Lin, F., & Sandkuhl, K. (2008). A Survey of Exploiting WordNet in Ontology Matching. In M. Bramer (Ed.), *Artificial Intelligence in Theory and Practice II* (pp. 341–350). Boston, MA: Springer US. doi:10.1007/978-0-387-09695-7\_33
- Madhavan, J., Jeffery, S. R., Cohen, S., Dong, X. (., Ko, D., Yu, C., & Halevy, A. (2007). Web-scale data integration: You can only afford to pay as you go. *Proceedings of CIDR 2007*, (pp. 342–350). Retrieved from <http://cidrdb.org/cidr2007/papers/cidr07p40.pdf>
- Mascardi, V., Locoro, A., & Rosso, P. (2010). Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 22(5), 609–623. doi:10.1109/TKDE.2009.154
- McCrae, J. P., & Buitelaar, P. (2018). Linking Datasets Using Semantic Textual Similarity. *Cybernetics and Information Technologies*, 18(1), 109 – 123. doi:10.2478/cait-2018-0010
- McCrae, J. P., Bosque-Gil, J., Gracia, J., Buitelaar, P., & Cimiano, P. (2017). The OntoLex-Lemon Model: Development and Applications. In I. Kosem, C. Tiberius, M. Jakubiček, J. Kallas, S. Krek, & V. Baisa (Eds.), *Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference.*, (pp. 587–597).
- McCrae, J., Aguado-de-Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gómez-Pérez, A., . . . Wunner, T. (2012). Interchanging lexical resources on the Semantic Web. *Language Resources and Evaluation*, 46(4), 701–719. Retrieved from <http://dx.doi.org/10.1007/s10579-012-9182-3>
- Mochol, M., & Jentzsch, A. (2008). Towards a Rule-Based Matcher Selection. In A. Gangemi, & J. Euzenat (Eds.),

- Knowledge Engineering: Practice and Patterns. EKAU 2008 (Lecture Notes in Computer Science)* (Vol. 5268, pp. 109-119). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-87696-0\_12
- Pazienza, M. T., Scarpato, N., Stellato, A., & Turbati, A. (2012). Semantic Turkey: A Browser-Integrated Environment for Knowledge Acquisition and Management. *Semantic Web Journal*, 3(3), 279-292. doi:10.3233/SW-2011-0033
- Pianta, E., Bentivogli, L., & Girardi, C. (2002). MultiWordNet: developing an aligned multilingual database. *Proceedings of the First international conference on global WordNet, Mysore, India , 21/01/2002 - 25/01/2002*, (pp. 293-302).
- Quix, C., Roy, P., & Kensche, D. (2011). Automatic Selection of Background Knowledge for Ontology Matching. In *Proceedings of the International Workshop on Semantic Web Information Management* (pp. 5:1--5:7). New York, NY, USA: ACM. doi:10.1145/1999299.1999304
- Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10, 334-350. doi:10.1007/s007780100057
- Röder, M., Kuchelev, D., & Ngonga Ngomo, A.-C. (2019). HOBbit: A platform for benchmarking Big Linked Data. *Data Science*. doi:10.3233/DS-190021
- Roventini, A., Alonge, A., Bertagna, F., Calzolari, N., Marinelli, R., Magnini, B., . . . Zampolli, A. (2002, January 21-25). ItalWordNet: A Large Semantic Database for the Automatic Treatment of the Italian Language. *First International WordNet Conference*. Mysore, India.
- Sabou, M., d'Aquin, M., & Motta, E. (2008). Exploring the Semantic Web as Background Knowledge for Ontology Matching. In S. Spaccapetra, J. Z. Pan, P. Thiran, T. Halpin, S. Staab, V. Svatek, . . . J. Roddick (Eds.), *Journal on Data Semantics XI (Lecture Notes in Computer Science)* (Vol. 5383, pp. 156-190). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-92148-6\_6
- Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3), 96-101. doi:10.1109/MIS.2006.62
- Shvaiko, P., & Euzenat, J. (2013, January). Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1), 158-176. doi:10.1109/TKDE.2011.253
- Stellato, A. (2015). A Language-Aware Web will Give Us a Bigger and Better Semantic Web. *Proceedings of the 4th Workshop on the Multilingual Semantic Web, co-located with the 12th Extended Semantic Web Conference - ESWC2015*. Portoroz, Slovenia.
- Stellato, A., Fiorelli, M., Turbati, A., Lorenzetti, T., van Gemert, W., Dechandon, D., . . . Keizer, J. (2020). VocBench 3: A collaborative Semantic Web editor for ontologies, thesauri and lexicons. *Semantic Web*, 11(5), 855-881. doi:10.3233/SW-200370
- Stellato, A., Rajbhandari, S., Turbati, A., Fiorelli, M., Caracciolo, C., Lorenzetti, T., . . . Pazienza, M. (2015). VocBench: a Web Application for Collaborative Development of Multilingual Thesauri. In F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, & A. Zimmermann (Eds.), *The Semantic Web. Latest Advances and New Domains (Lecture Notes in Computer Science)* (Vol. 9088, pp. 38-53). Springer, Cham. doi:10.1007/978-3-319-18818-8\_3
- Tartir, S., & Arpinar, I. B. (2007). Ontology Evaluation and Ranking using OntoQA. In *International Conference on Semantic Computing (ICSC 2007)* (pp. 185-192). IEEE. doi:10.1109/ICSC.2007.19
- Wiederhold, G. (1994). Interoperation, Mediation and Ontologies. *Proceedings International Symposium on Fifth Generation Computer Systems (FGCS94), Workshop on Heterogeneous Cooperative Knowledge Bases* (pp. 33-48). Tokio, Japan: ICOT.
- World Wide Web Consortium (W3C). (2014, January 16). *Data Catalog Vocabulary (DCAT)*. (F. Maali, & J. Erickson, Eds.) Retrieved from World Wide Web Consortium (W3C): <http://www.w3.org/TR/vocab-dcat/>



**Figure 1** Overall architecture of the framework MAPLE. The picture shows dependencies (dashed rectangles) and how components interact (solid arrows).



**Figure 2** Metadata about the EuroVoc thesaurus.

```

1. :EuroVoc a void:Dataset ;
2.   dcterms:title "EuroVoc";
3.   dcterms:description "EuroVoc is a multilingual, multidisciplinary thesaurus covering the activities of the EU. [..]"@en;
4.   void:sparqlEndpoint <http://localhost:7200/repositories/EuroVoc>;
5.   void:uriSpace "http://eurovoc.europa.eu/";
6.   dcterms:conformsTo <http://www.w3.org/2004/02/skos/core>;
7.   void:classPartition [
8.     void:class <http://www.w3.org/2004/02/skos/core#ConceptScheme>;
9.     void:entities 130
10.  ], [
11.    void:class <http://www.w3.org/2004/02/skos/core#Concept>;
12.    void:entities 7154 .
13.  ], [
14.    void:class <http://www.w3.org/2004/02/skos/core#Collection>;
15.    void:entities 0 .
16.  ],
17. void:triples 2157673;
18. void:entities 7284;
19. void:distinctSubjects 398289;
20. void:distinctObjects 826890;
21. void:subset :EuroVoc_en_lexicalization_set, :EuroVoc_OP_AT_Country_linkset, [...]
22. .
23. :EuroVoc_en_lexicalization_set a lime:LexicalizationSet;
24. lime:referenceDataset :EuroVoc;
25. dcterms:language <http://id.loc.gov/vocabulary/iso639-1/en>, <http://lexvo.org/id/iso639-3/eng>;
26. lime:language "en";
27. lime:lexicalizationModel <http://www.w3.org/2008/05/skos-xl>;
28. lime:lexicalizations 16447;
29. lime:references 7284
30. lime:avgNumOfLexicalizations 2.258;
31. lime:percentage 1.0;
32. .
33. [...]
34. :EuroVoc_OP_AT_Country_linkset a void:Linkset ;
35.   void:subjectsTarget :EuroVoc ;
36.   void:objectsTarget :OP_AT_Country ;
37.   void:linkPredicate <http://www.w3.org/2002/07/owl#sameAs> ;
38.   void:triples 247
39.   [...]
40. .

```

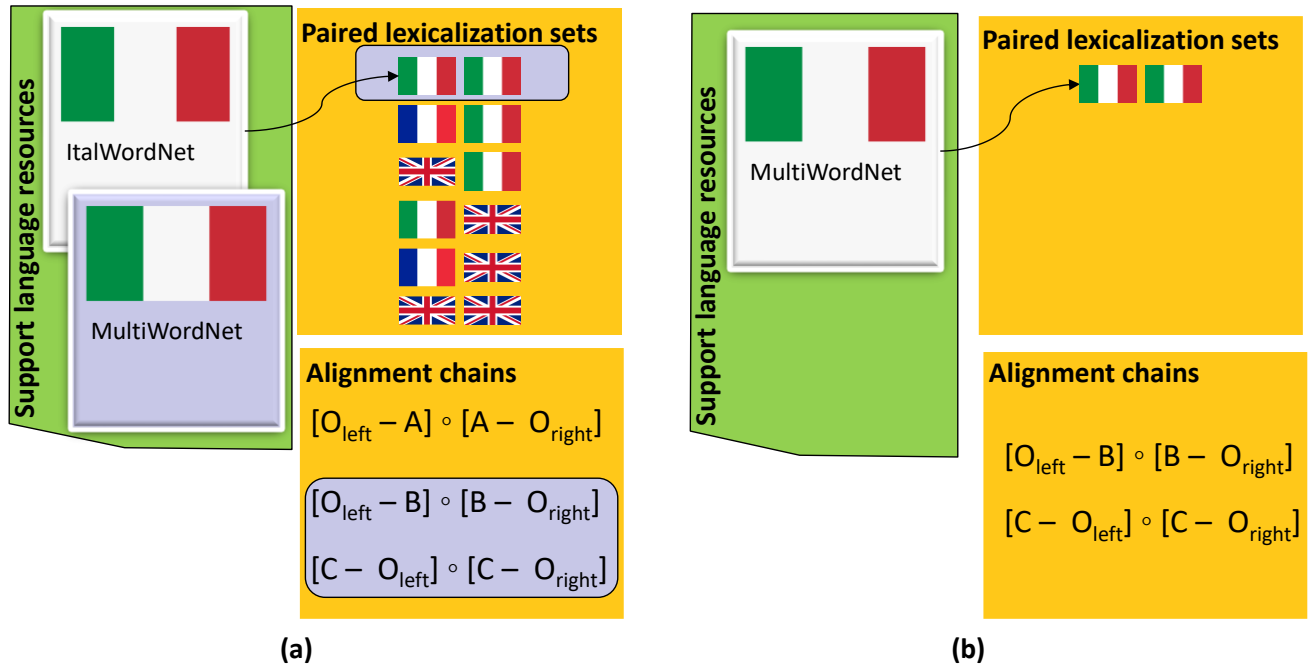
**Figure 3** Metadata about Open Multilingual Wordnet.

```

1. :OMW a void:Dataset;
2.   void:uriSpace "http://art.uniroma2.it/pmki/omw/";
3.   dcterms:title "Open Multilingual Wordnet"@en;
4.   dcterms:description "It provides access to open wordnets in a variety of languages, all linked to the
   Princeton Wordnet of English (PWN)"@en;
5.   void:sparqlEndpoint <http://localhost:7200/repositories/OMW_core>;
6.   dcterms:conformsTo <http://www.w3.org/2004/02/skos/core>;
7.   void:classPartition [
8.       void:class <http://www.w3.org/2004/02/skos/core#Concept>;
9.       void:entities 117659
10.  ];
11.  void:classPartition [
12.      void:class <http://www.w3.org/2004/02/skos/core#Collection>;
13.      void:entities 0
14.  ];
15.  void:classPartition [
16.      void:class <http://www.w3.org/2004/02/skos/core#ConceptScheme>;
17.      void:entities 1
18.  ];
19.  void:triples 18654333;
20.  void:entities 117660;
21.  void:distinctSubjects 5030850;
22.  void:distinctObjects 6208908;
23.  void:subset <http://art.uniroma2.it/pmki/omw/pwn30-conceptset>,
24.              <http://art.uniroma2.it/pmki/omw/Princeton_WordNet-en-lexicon>, [...]
              <http://art.uniroma2.it/pmki/omw/WOLF_(Wordnet_Libre_du_Fran%C3%A7ais)-fr-lexicon>;
25. .
26. <http://art.uniroma2.it/pmki/omw/pwn30-conceptset> a ontolex:ConceptSet;
27.   lime:concepts 117659 .
28. <http://art.uniroma2.it/pmki/omw/Princeton_WordNet-en-lexicon> a lime:Lexicon;
29.   dcterms:language <http://id.loc.gov/vocabulary/iso639-1/en>, <http://lexvo.org/id/iso639-3/eng>;
30.   lime:language "en";
31.   lime:lexicalEntries 156584
32. .
33. :Princeton_WordNet-en-lexicon_pwn30-conceptset_conceptualization_set
34.   a lime:ConceptualizationSet;
35.   lime:conceptualizations 206978;
36.   lime:concepts 117659;
37.   lime:lexicalEntries 156584;
38.   lime:avgAmbiguity 1.322;
39.   lime:avgSynonymy 1.76;
40.   lime:conceptualDataset <http://art.uniroma2.it/pmki/omw/pwn30-conceptset>;
41.   lime:lexiconDataset <http://art.uniroma2.it/pmki/omw/Princeton_WordNet-en-lexicon>
42. .

```

**Figure 4** An alignment scenario (a) with highlighted the choices that will ultimately lead to the corresponding scenario definition (b).



**Figure 5** Alignment scenario of TESEO and EuroVoc.

```

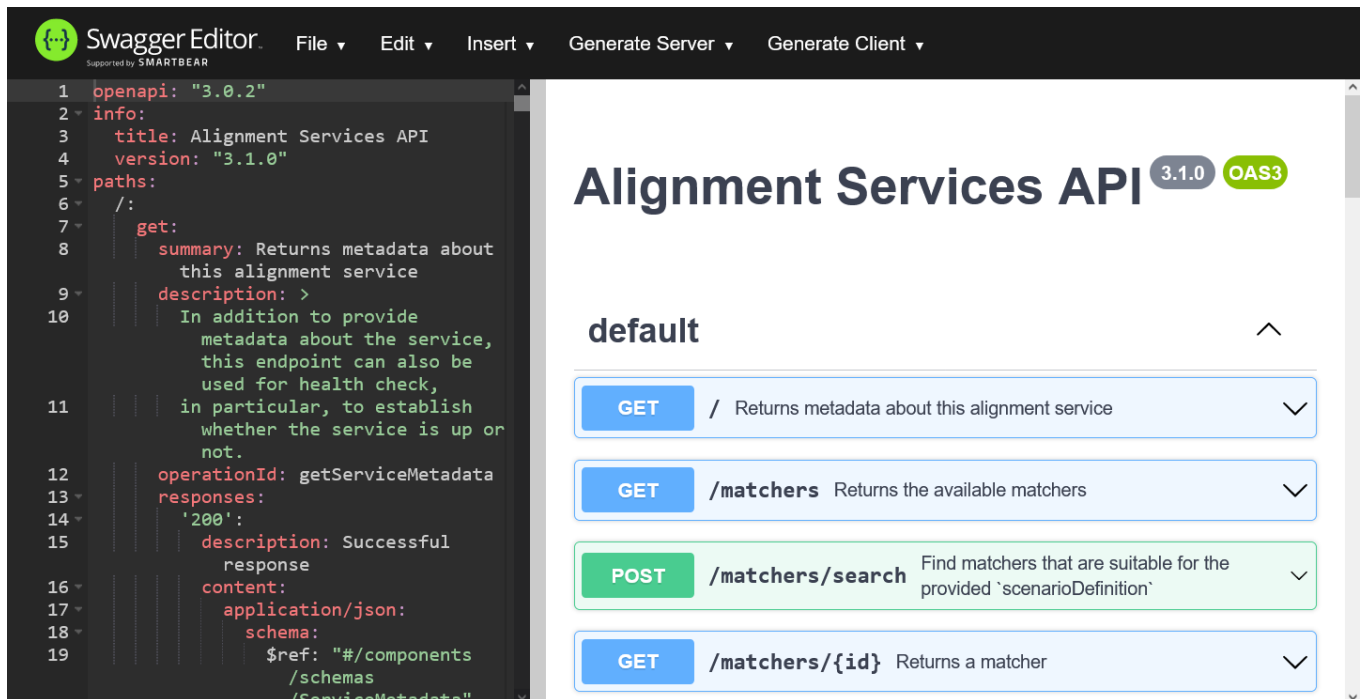
1.  {
2.    "result": {
3.      "leftDataset": {
4.        "@id": "http://semanticturkey.uniroma2.it/metadataregistry/bb236470-b49d-4da1-bd9d-ba475fcd194c",
5.        "@type": "http://rdfs.org/ns/void#Dataset",
6.        "uriSpace": "http://www.senato.it/teseo/tes/",
7.        "title": [ "TESEO" ],
8.        "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/TESEO_core", "username": null, "password": null },
9.        "conformsTo": "http://www.w3.org/2004/02/skos/core"
10.     },
11.     "rightDataset": {
12.       "@id": "http://semanticturkey.uniroma2.it/metadataregistry/43789b48-8165-404d-a9c7-ac251da44e71",
13.       "@type": "http://rdfs.org/ns/void#Dataset",
14.       "uriSpace": "http://eurovoc.europa.eu/",
15.       "title": [ "EuroVoc" ],
16.       "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/EuroVoc_core", "username": null, "password": null },
17.       "conformsTo": "http://www.w3.org/2004/02/skos/core"
18.     },
19.     "supportDatasets": [
20.       {
21.         "@id": "http://art.uniroma2.it/pmki/omw/ItalWordnet-it-lexicon", "@type": "http://www.w3.org/ns/lemon/lime#Lexicon",
22.         "title": [ "ItalWordnet" ],
23.         "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/OMW_core", "username": null, "password": null },
24.         "languageTag": "it",
25.         "lexicalEntries": 19680
26.       }, {
27.         "@id": "http://art.uniroma2.it/pmki/omw/MultiWordNet-it-lexicon", "@type": "http://www.w3.org/ns/lemon/lime#Lexicon",
28.         "title": [ "MultiWordNet" ],
29.         "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/OMW_core", "username": null, "password": null },
30.         "languageTag": "it",
31.         "lexicalEntries": 43011
32.       }, {
33.         "@id": "http://art.uniroma2.it/pmki/omw/pwn30-conceptset", "@type": "http://www.w3.org/ns/lemon/ontolex#ConceptSet",
34.         "title": [ { "@value": "Princeton WordNet 3.0 Concept Set", "@lang": "en" } ],
35.         "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/OMW_core", "username": null, "password": null },
36.         "concepts": 117659
37.       }, {
38.         "@id": "http://semanticturkey.uniroma2.it/metadataregistry/067a0a4e-6a5c-4a61-8776-2583603cfdad",
39.         "@type": "http://rdfs.org/ns/void#Dataset",
40.         "uriSpace": "http://www.eionet.europa.eu/gemet/",
41.         "title": [ "GEMET" ],
42.         "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/GEMET_core", "username": null, "password": null },
43.         "conformsTo": "http://www.w3.org/2004/02/skos/core"
44.       }, {
45.         "@id": "http://semanticturkey.uniroma2.it/metadataregistry/EuroVoc_it_lexicalization_set",
46.         "title": [],
47.         "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/EuroVoc_core", "username": null, "password": null },
48.         "referenceDataset": "http://semanticturkey.uniroma2.it/metadataregistry/43789b48-8165-404d-a9c7-ac251da44e71",
49.         "lexiconDataset": null,
50.         "lexicalizationModel": "http://www.w3.org/2008/05/skos-xl",
51.         "lexicalizations": 18545, "references": 7282, "lexicalEntries": null, "avgNumOfLexicalizations": 2.546, "percentage": 1.0,
52.         "languageTag": "it",
53.         "@type": "http://www.w3.org/ns/lemon/lime#LexicalizationSet"
54.       }, {
55.         "@id": "http://semanticturkey.uniroma2.it/metadataregistry/ItalWordnet-it-lexicon_pwn30-conceptset_conceptualization_set",
56.         "@type": "http://www.w3.org/ns/lemon/lime#ConceptualizationSet",
57.         "uriSpace": null,
58.         "title": [],
59.         "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/OMW_core", "username": null, "password": null },
60.         "lexiconDataset": "http://art.uniroma2.it/pmki/omw/ItalWordnet-it-lexicon",
61.         "conceptualDataset": "http://art.uniroma2.it/pmki/omw/pwn30-conceptset",
62.         "conceptualizations": 24135, "concepts": 15563, "lexicalEntries": 19680, "avgSynonymy": 0.206, "avgAmbiguity": 1.227
63.       },
64.     ],
65.     "@id": "http://semanticturkey.uniroma2.it/metadataregistry/MultiWordNet-it-lexicon_pwn30-conceptset_conceptualization_set",
66.     "@type": "http://www.w3.org/ns/lemon/lime#ConceptualizationSet",

```

```

67.   "uriSpace": null,
68.   "title": [],
69.   "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/OMW_core", "username": null, "password": null },
70.   "lexiconDataset": "http://art.uniroma2.it/pmki/omw/MultiWordNet-it-lexicon",
71.   "conceptualDataset": "http://art.uniroma2.it/pmki/omw/pwn30-conceptset",
72.   "conceptualizations": 63133, "concepts": 35001, "lexicalEntries": 43011, "avgSynonymy": 0.537, "avgAmbiguity": 1.468
73. }, {
74.   "@id": "http://semanticturkey.uniroma2.it/metadataregistry/TESEO_it_lexicalization_set",
75.   "uriSpace": null,
76.   "title": [],
77.   "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/TESEO_core", "username": null, "password": null },
78.   "referenceDataset": "http://semanticturkey.uniroma2.it/metadataregistry/bb236470-b49d-4da1-bd9d-ba475fcd194c",
79.   "lexiconDataset": null,
80.   "lexicalizationModel": "http://www.w3.org/2008/05/skos-xl",
81.   "lexicalizations": 3378, "references": 3378, "lexicalEntries": null, "avgNumOfLexicalizations": 1.0, "percentage": 1.0,
82.   "languageTag": "it",
83.   "@type": "http://www.w3.org/ns/lemon/leme#LexicalizationSet"
84. }, {
85.   "@id": "urn:uuid:a5ddd5ce-137d-4ff2-8a73-f7de450c0ca1",
86.   "uriSpace": null,
87.   "title": [],
88.   "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/TESEO_GEMET ", "username": null, "password": null },
89.   "subjectsTarget": "http://semanticturkey.uniroma2.it/metadataregistry/bb236470-b49d-4da1-bd9d-ba475fcd194c",
90.   "objectsTarget": "http://semanticturkey.uniroma2.it/metadataregistry/067a0a4e-6a5c-4a61-8776-2583603cfdad",
91.   "@type": "http://semanticturkey.uniroma2.it/ns/mdr#Alignment"
92. }, {
93.   "@id": "urn:uuid:cc10aaef-3441-4f02-855e-5f5d93a69ce7",
94.   "uriSpace": null,
95.   "title": [],
96.   "sparqlEndpoint": { "endpointURL": "http://localhost:7200/repositories/GEMET_core", "username": null, "password": null },
97.   "subjectsTarget": "http://semanticturkey.uniroma2.it/metadataregistry/067a0a4e-6a5c-4a61-8776-2583603cfdad",
98.   "objectsTarget": "http://semanticturkey.uniroma2.it/metadataregistry/43789b48-8165-404d-a9c7-ac251da44e71",
99.   "@type": "http://semanticturkey.uniroma2.it/ns/mdr#Alignment"
100. }
101. ],
102. "pairings": [
103.   {
104.     "score": 0.5716210939615214,
105.     "bestCombinedScore": 0.7836831074710862,
106.     "source": {
107.       "lexicalizationSet": "http://semanticturkey.uniroma2.it/metadataregistry/TESEO_it_lexicalization_set"
108.     },
109.     "target": {
110.       "lexicalizationSet": "http://semanticturkey.uniroma2.it/metadataregistry/EuroVoc_it_lexicalization_set"
111.     },
112.     "synonymizers": [
113.       {
114.         "score": 0.7836831074710862,
115.         "lexicon": "http://art.uniroma2.it/pmki/omw/MultiWordNet-it-lexicon",
116.         "conceptualizationSet": "http://semanticturkey.uniroma2.it/metadataregistry/MultiWordNet-it-lexicon_pwn30"
117.       }, {
118.         "score": 0.606037008877326,
119.         "lexicon": "http://art.uniroma2.it/pmki/omw/ItalWordnet-it-lexicon",
120.         "conceptualizationSet": "http://semanticturkey.uniroma2.it/metadataregistry/ItalWordnet-it-lexicon_pwn30"
121.       }
122.     ]
123.   },
124. ],
125. "alignmentChains": [
126.   { "score": 1.0, "chain": [ "urn:uuid:a5ddd5ce-137d-4ff2-8a73-f7de450c0ca1", "urn:uuid:cc10aaef-3441-4f02-855e-5f5d93a69ce7" ] }
127. ]
128. }
129. }

```

**Figure 6** The OpenAPI specification of the remote alignment services API shown inside the online Swagger Editor.

The image shows the Swagger Editor interface for the 'Alignment Services API'. The left pane displays the OpenAPI specification in JSON format, and the right pane shows a visual representation of the API endpoints.

**OpenAPI Specification (Left Pane):**

```
1 openapi: "3.0.2"
2 info:
3   title: Alignment Services API
4   version: "3.1.0"
5 paths:
6   /:
7     get:
8       summary: Returns metadata about
9       this alignment service
10      description: >
11        In addition to provide
12        metadata about the service,
13        this endpoint can also be
14        used for health check,
15        in particular, to establish
16        whether the service is up or
17        not.
18      operationId: getServiceMetadata
19      responses:
20        '200':
21          description: Successful
22          response
23          content:
24            application/json:
25              schema:
26                $ref: "#/components
27                /schemas
28                /ServiceMetadata"
```

**Visual Representation (Right Pane):**

**Alignment Services API** 3.1.0 OAS3

**default**

- GET** / Returns metadata about this alignment service
- GET** /matchers Returns the available matchers
- POST** /matchers/search Find matchers that are suitable for the provided 'scenarioDefinition'
- GET** /matchers/{id} Returns a matcher

**Figure 7** Submission of an alignment task to a remote alignment service in VocBench 3.

### Create task

Left project

TESEO

✓

Right project

EuroVoc

✓

Profile matching

Type: Dataset

URI space: <http://www.senato.it/teseo/tes/>

Conforms to: SKOS

SPARQL Endpoint: [http://localhost:7200/repositories/TESEO\\_core](http://localhost:7200/repositories/TESEO_core)

Type: Dataset

URI space: <http://eurovoc.europa.eu/>

Conforms to: SKOS

SPARQL Endpoint: [http://localhost:7200/repositories/EuroVoc\\_core](http://localhost:7200/repositories/EuroVoc_core)

#### Pairings

☒ Use

Score:0.572

Best combined score:0.784

Synonymizers ⓘ

☐

MultiWordNet

Score: 0.784 ⓘ

☐

ItalWordnet

Score: 0.606 ⓘ

#### Matchers

Optionally a matcher can be provided to the alignment system. Click [here](#) to search for available matchers.

Ok

Cancel



**Figure 8** Alignment validation panel in VocBench 3.

**Alignment Validation:**

Source: Genoma task

**Tasks** + ↺

Left	Right	Status	Start time	End time	
Teseo	Eurovoc	Completed	Mon Jun 03 10:49:31 +0000 2019	Mon Jun 03 10:50:06 +0000 2019	<a href="#">edit alignment</a>

**Alignments:** A Settings

FUNZIONI DI SPESA (it)	administrative expenditure (EU) (en), spesa di funzionamento (UE) (it)	<div><div></div><div></div></div> = <div><div></div><div></div></div>	<a href="#">Accept</a> <a href="#">Reject</a>
OMOLOGAZIONE DI PRODOTTI (it)	product designation (en), denominazione del prodotto (it)	<div><div></div><div></div></div> = <div><div></div><div></div></div>	<a href="#">Accept</a> <a href="#">Reject</a>
ASSEGNO E DOTAZIONE DEL PRESIDENTE DELLA REPUBBLICA (it)	product designation (en), denominazione del prodotto (it)	<div><div></div><div></div></div> = <div><div></div><div></div></div>	<a href="#">Accept</a> <a href="#">Reject</a>
TRASFERIMENTO DI PERSONALE (it)	Stabex (en), Stabex (it)	<div><div></div><div></div></div> = <div><div></div><div></div></div>	<a href="#">Accept</a> <a href="#">Reject</a>
CREDITO AGRARIO DI FUNZIONAMENTO (it)	administrative expenditure (EU) (en), spesa di funzionamento (UE) (it)	<div><div></div><div></div></div> = <div><div></div><div></div></div>	<a href="#">Accept</a> <a href="#">Reject</a>
TRASFERIMENTO DI ABITATI (it)	Stabex (en), Stabex (it)	<div><div></div><div></div></div> = <div><div></div><div></div></div>	<a href="#">Accept</a> <a href="#">Reject</a>

**Quick Actions:** --- Do quick action [Apply to Ontology](#) [Export as...](#)