

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/320899408>

# YAM++ Online: A Web Platform for Ontology and Thesaurus Matching and Mapping Validation

**Conference Paper** in Lecture Notes in Computer Science · November 2017

DOI: 10.1007/978-3-319-70407-4\_26

CITATIONS

3

READS

52

4 authors, including:



**Zohra Bellahsene**

Université de Montpellier

148 PUBLICATIONS 1,147 CITATIONS

[SEE PROFILE](#)



**Vincent Emonet**

Maastricht University

14 PUBLICATIONS 58 CITATIONS

[SEE PROFILE](#)



**Duyhoa Ngo**

Université de Montpellier

21 PUBLICATIONS 256 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



AgroPortal [View project](#)



Semantic Indexing of French Biomedical Data Resources (SIFR) [View project](#)

# YAM++ *Online*: A Web Platform for Ontology and Thesaurus Matching and Mapping Validation

Zohra Bellahsene<sup>(✉)</sup>, Vincent Emonet, Duyhoa Ngo, and Konstantin Todorov

University of Montpellier, LIRMM, Montpellier, France  
{bella,todorov}@lirmm.fr

**Abstract.** We present the multi-task web platform YAM++ *online* for ontology and thesaurus matching, featuring a mapping validation and enrichment interface. The online matcher is based on the YAM++ system. The validator allows to visualize an alignment, edit the relation type and add new mappings discovered through a keyword-based search by a domain expert.

## 1 Introduction

Ontology matching, or alignment, is the process of (semi-) automatically discovering correspondences between entities belonging to different ontologies. Several approaches have been proposed and many matching systems have been designed in the past years [4]. However, few of them provide a user-friendly and easy to access interface that allows domain experts, having only basic technical knowledge of the semantic web field and the matching process per se, to execute and manipulate the alignments [1, 2]. To take two examples, VOAR<sup>1</sup> allows to manually produce mappings or manipulate existing ones, whereas LogMap<sup>2</sup> allows to send a mapping request via a web platform, while the result (the alignment) is sent by email back to the user. In contrast, YAM++ *online*<sup>3</sup>, provides an intuitive web environment for both the online execution *and* validation of ontology matching tasks, allowing for (not requiring) the active user participation.

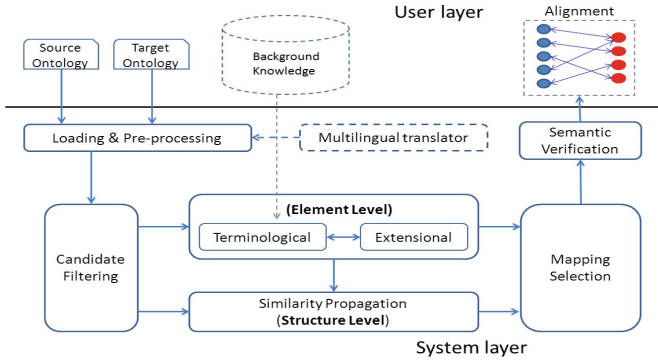
## 2 Overview of the Platform

**Matcher.** We present concisely the main features of the YAM++ system [3], evaluated on several OAEI benchmarks in the past years, underlying the matcher module of YAM++ *online*. YAM++ ((not) Yet Another Matcher) is a multi-strategy and self-configuring system for discovering equivalence relations between ontology elements (classes, object properties and datatype properties). Figure 1 presents its architecture, described briefly below.

<sup>1</sup> <http://voar.inf.pucrs.br>.

<sup>2</sup> <http://krrwebtools.cs.ox.ac.uk/logmap/>.

<sup>3</sup> <http://yamplusplus.lirmm.fr>.



**Fig. 1.** Main components of YAM++.

Input ontologies are loaded and parsed by the *Ontology Loader* component; the ontology entities, according to their types, are indexed by the *Annotation Indexing*, the *Structure Indexing* and *Context Indexing* components; the *Candidates Pre-Filtering* component filters out all pairs of entities with highly similar descriptions; the candidate mappings are then passed into the *Similarity Computation* component, which includes: (i) a *Terminological Matcher* that produces a set of mappings by comparing the annotations of entities; (ii) an *Instance-based Matcher* that supplements new mappings through shared instances between ontologies and (iii) a *Contextual Matcher*, which is used to compute the similarity value of a pair of entities by comparing their context profiles. In YAM++, the matching results of the *Terminological Matcher*, the *Contextual Matcher* and the *Instance-based Matcher* are combined to produce a single merged set of mappings. The *Similarity Propagation* component then enhances the element level matching result by exploiting the structure of the ontologies. The *Candidate Post-Filtering* component is used to combine and select the potential candidate mappings from the element and the structural level results. Finally, the *Semantic Verification* component checks the consistency of the discovered mappings.

**HTTP API.** YAM++ *online* provides a way to perform the ontology matching programmatically by submitting the two ontology files through a HTTP request. Local files can be submitted using a HTTP POST request, while ontologies available on the web, through an URL, can be submitted using a HTTP GET request. Using the API requires an API key that is provided after creating an account. The HTTP response is provided using the alignment results in the XML EDOAL alignment format. The HTTP API allows users to easily integrate the matcher in their programs or workflow without having to install the YAM++ library. However, in case of an intensive use of the matcher, it is recommended to directly use the YAM++ library available on Maven.

**Validator.** An alignment produced by YAM++, or any other automatic system, can potentially contain erroneous mappings, or be incomplete. Therefore, the user participation at the mapping validation phase occurs to be important in

real life situations, where the matching system will be seen as a generator of a (hopefully) significant in size and quality pool of candidates and the domain expert will have the final say. By domain here we mean the area of life that the aligned ontologies describe (e.g., biomedicine or music). The Validator module of YAM++ *online* allows for both the manual expert validation *and* enrichment of the automatically generated mappings.

### 3 Demonstration Scenario

We will go through the different components of the web interface of YAM++ *online*.

The screenshot shows the YAM++ online matcher interface. At the top, a dropdown menu is set to 'OAEI Large Bio (FMA - SNOMED)'. Below it, a yellow box contains a warning: 'SKOS scheme are converted to OWL (currently supported by YAM++), so mind that the semantics may be slightly altered (skos:broader and skos:narrower to owl:subClassOf). OBO format is currently not supported'. The interface is split into two columns: 'Source' and 'Target'. Each column has a 'Source file' or 'Target file' input field with a URL 'https://git.lirmm.fr/opendata/yampp-is/r/' and a 'Choose file' button. Below each input field is a 'Source ontology name:' or 'Target ontology name:' label with a text box. A 'Hide matcher parameters' button is centered below these fields. At the bottom, there are two panels. The left panel, 'Matcher parameters', has a green header and contains a checked checkbox 'Match synonyms to synonyms' with a note: 'By default Yam++ matches the preferred label (e.g.: skos:prefLabel) to the preferred label, and the synonyms (e.g.: skos:altLabel) to the'. The right panel, 'Remove conflicts', has a yellow header and contains a note: 'Enabling the removal of conflicts decreases the number of mappings, but also the likelihood of error (wrong mappings).' and an unchecked checkbox 'Remove Fxfirst Fxlast conflicts'.

**Fig. 2.** The matcher GUI with the option “Show matcher parameters” on.

After login, **the Matcher** opens with a page where one can upload two ontologies to map by using a file path or a url, in one of the indicated input formats covering valid OWL and SKOS files, as shown in Fig. 2. A set of pairs of real-world ontologies are available for testing, including two medium-size pairs of SKOS vocabularies dedicated to musical instruments (IAML with 419 terms and DIABOLO – with 2117) and genres (REDOMI with 313 terms and RAMEAU – with 654)<sup>4</sup>, respectively, issued from the DOREMUS project<sup>5</sup>. Further on the list, one finds pairs of (fragments of) larger ontologies from the OAEI campaign<sup>6</sup>: the

<sup>4</sup> <https://github.com/DOREMUS-ANR/knowledge-base>.

<sup>5</sup> <http://www.doremus.org>.

<sup>6</sup> [www.oaei.org](http://www.oaei.org).

well-known *Anatomy* track ontologies (*Adult Mouse Anatomy* with 2744 concepts and a fragment of NCI with 3304 concepts) and three *Large bio* tasks ontologies (FMA with 10157 classes, SNOMED with 13412 and a larger fragment of NCI with 6488 concepts).

Before launching the execution, the user is able to determine whether the matching will be based on comparing only `skos:prefLabel` values or also synonyms from the `skos:altLabel` properties (Fig. 2, bottom left). The option of managing conflicts allows to improve recall (Fig. 2, bottom right). Once YAM++ has executed, the result is displayed on the Validator page (Fig. 3).

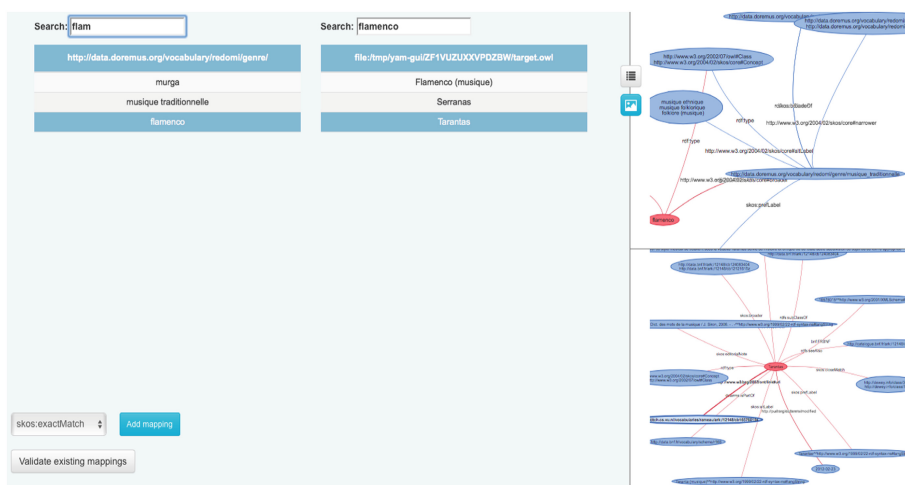
The screenshot displays the Validator GUI. At the top, three status boxes show: 'Running time' (16.872 seconds), 'Source ontology mapped' (Matched 116 entities on 296, 39%), and 'Target ontology mapped' (Matched 116 entities on 663, 17%). Below these is a search bar and a 'Show scores from 0.50 to 0.70' slider. The main table lists mappings between two ontologies. A dropdown menu is open for the 'nocturne (musique)' vs 'Nocturnes (musique)' pair, showing options like 'skos:exactMatch' (checked), 'skos:closeMatch', 'skos:broadMatch', 'skos:narrowMatch', 'skos:relatedMatch', 'Not valid', and 'skos:exactMatch'. To the right, 'Source entity details' for 'nocturne (musique)' and 'Target entity details' for 'Nocturnes (musique)' are shown, including their URIs and descriptions.

Source	Target	Relation	Score
biguine	Biguines	skos:exactMatch	0.51
sonate	Sonates	skos:exactMatch	0.55
gigue	Gigues	skos:exactMatch	0.51
nocturne (musique)	Nocturnes (musique)	skos:exactMatch	0.61
bagatelle (musique)	Bagatelles (musique)	skos:exactMatch	0.58
nouba	Noubas	skos:exactMatch	0.51
acid jazz	Acid jazz	skos:exactMatch	0.67
	Tarentelles	skos:exactMatch	0.57

**Fig. 3.** The validator GUI. A text-based representation of the selected match pair on the right.

Note that the **the Validator interface** can be seen as a standalone tool, because it takes as an input a valid EDOAL alignment that is not necessarily produced by YAM++, together with its two ontologies (via an URL or a file path). A list of mappings (pairs of labels of aligned concepts) appears on the main page, together with information about the portions of the ontologies covered by the alignment (Fig. 3, above). A context description of each of the two concepts in each line is displayed (Fig. 3, right), containing all alternative labels, as well as the labels (or URIs) of parents and children. One can choose between a textual description (reaching a perimeter of distance 1 around the concept of interest) and a graphical visualization (reaching a distance 2). The confidence score, if available in the initial alignment, is provided for each pair of concepts (at the right end of each line) and the display of mappings can be filtered with respect to this score by the help of a horizontal cursor. For each concept pair, the expert is given the possibility to select a relation type from a list of SKOS relations (`skos:exactMatch` being the default choice), or simply discard the matching.

In case the expert feels that the provided list of mappings is incomplete or wants to look for a correct match to replace a mapping that they have invalidated, they can proceed to the alignment enrichment environment, by clicking on the “Add new mappings” button (Fig. 3, bottom left). A new page opens containing the full concept label lists of the two ontologies (Fig. 4). A key-word search on both lists, including preferred and alternative labels, allows to browse and select manually a pair of concepts and define their relation. Once done, the newly defined mappings are added to the initial alignment. Figure 4 also shows the alternative graphical visualization of the source and target concepts contexts.



**Fig. 4.** Adding manually new mappings. A graph-based alternative representation of the selected classes contexts is seen on the right.

Finally, all modifications are added to the alignment file, which can be either saved in the default EDOAL internal format, or exported in the form of simple triples.

## 4 Future Work

In the future, we plan to extend the functionalities of YAM++ *online* by allowing for ontology recommendation, use and selection of background knowledge for a given matching task, alignment validation by crowdsourcing, as well as adding an interface for linkset validation in an instance matching scenario.

**Acknowledgements.** This work has been partially supported by the French National Research Agency (ANR) within the DOREMUS Project, under grant number ANR-14-CE24-0020.

## References

1. Aumuellner, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: SIGMOD, pp. 906–908 (2005)
2. Dragisic, Z., Ivanova, V., Lambrix, P., Faria, D., Jiménez-Ruiz, E., Pesquita, C.: User validation in ontology alignment. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9981, pp. 200–217. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46523-4\\_13](https://doi.org/10.1007/978-3-319-46523-4_13)
3. Ngo, D.H., Bellahsene, Z.: Overview of YAM++ - (not) yet another matcher for ontology alignment task. *J. Web Sem.* **41**, 30–49 (2016)
4. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. *IEEE Trans. Knowl. Data Eng.* **99**, 158–176 (2013)