

Article

Alignment: A Hybrid, Interactive and Collaborative Ontology and Entity Matching Service [†]

Sotirios Karampatakis ^{1,2,*} , Charalampos Bratsas ^{1,2} , Ondřej Zamazal ³ ,
Panagiotis Marios Filippidis ¹ and Ioannis Antoniou ²

¹ Open Knowledge Greece, 54352 Thessaloniki, Greece; cbratsas@math.auth.gr (C.B.); pafilipp@jour.auth.gr (P.M.F.)

² School of Mathematics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; iantonio@math.auth.gr

³ Department of Information and Knowledge Engineering, University of Economics, 13067 Prague, Czech Republic; ondrej.zamazal@vse.cz

* Correspondence: sokaramp@auth.gr

[†] This manuscript is an extended version of our paper “Alignment: A collaborative, system aided, interactive ontology matching platform” published in the Proceedings of Knowledge Engineering and Semantic Web, Szczecin, Poland, 9–10 November 2017.

Received: 9 October 2018; Accepted: 12 November 2018; Published: 15 November 2018



Abstract: Ontology matching is an essential problem in the world of Semantic Web and other distributed, open world applications. Heterogeneity occurs as a result of diversity in tools, knowledge, habits, language, interests and usually the level of detail. Automated applications have been developed, implementing diverse aligning techniques and similarity measures, with outstanding performance. However, there are use cases where automated linking fails and there must be involvement of the human factor in order to create, or not create, a link. In this paper we present Alignment, a collaborative, system aided, interactive ontology matching platform. Alignment offers a user-friendly environment for matching two ontologies with the aid of configurable similarity algorithms.

Keywords: linked data; ontology matching; SKOS; thesauri

1. Introduction

The Web of Data is an ever growing and massive amount of data from a wide range of domains such as publications, geographic, information, economic, health and agriculture among others that have become available. In fact, the diversity in data from distinct domains but also within the same or similar domains has created the need for linking and integrating data from numerous sources. Semantic Web technologies and the RDF format have tended to become a lingua franca of data integration [1]. Additionally, one of the most important features Semantic Web has to offer is its ability to interconnect data in order to get additional information. Thus, innovative and advanced approaches to link discovery in semantic resources can offer great possibilities and significantly benefit the integration task of diverse data.

While there is a variety of tools for automated or semi-automated link discovery, there are cases where this approach is either insufficient or inefficient. The potential complexity and versatility of vocabularies could easily lead to incorrect links. At the same time, the appearance of many synonyms of a particular vocabulary term in another vocabulary causes a common problem for creating accurate links between them [2]. An automatic approach of linking large vocabularies leads to only partial alignments amongst them, and along with this, configuration of similarity algorithms may produce no links for a number of entities, or include false positive links [3].

Consequently, linking ontologies can be a challenging and in some cases an unfeasible task to accomplish, by following a single approach. Moreover, there are cases where the best algorithm for finding links between entities is human knowledge [4]. This is especially true when the quality of mappings has a higher value than the quantity of links, which is the case for ontology matching. So, a mixed-method approach is probably the most suitable for this complicated task.

To this end, we have developed the Alignment Platform, a hybrid, interactive and collaborative ontology and entity matching service. The service is realized as an online platform where multiple users can co-create and validate linksets, or deployed locally as a standalone application.

This paper is an extended and improved version of [5]. The following sections and information have been added:

- a section presenting our study on similarity algorithms, explaining our decision on the default configuration of the suggestions engine
- a section presenting two use cases
- a section presenting related work
- updated text within existing sections

In Section 2 we present previous work on applications with related functionality with the Alignment Platform. In Section 3 we present a typical workflow using the platform. Additionally, we present the main components of the platform, outlining key functionalities of each component. In Section 3.3 we present our study on the string similarity algorithms used on the platform. Continuing, in Section 4, the three evaluation tests we conducted are described in detail. Finally, in Section 4.3 a description of two use cases where the Alignment Platform was used is provided. An online public instance of the platform is available on <https://alignment.okfn.gr> and the source code is available under the MIT license (<https://github.com/okgreece/Alignment>) through Github. A Docker image is provided as well through the Docker Hub (<https://hub.docker.com/r/skarampatakis/alignment/>).

2. Related Work

The Alignment platform is related to other works in different aspects. VocBench [6] is an online environment aimed at a collaborative development of OWL ontologies, SKOS thesauri and RDF datasets. Although it is mainly concentrated on development of thesauri, code lists and authority resources, it also enables a user to validate ontology alignments. In VocBench the user can upload existing alignments in the Alignment API format [7] and revise them. In comparison with the Alignment platform, the user cannot manually create links between ontologies/vocabularies in a user friendly environment. The user can manually create a link only by accessing the Data tab on the VocBench Frontend and adding the link as a property to the entity.

There are several related online tools from ontology matching field. The YAM++ Online is an online ontology and thesaurus matching tool. It is the online version of the YAM++ tool [8], with the difference that the user does not need to deploy the service locally. The YAM++ Online tool offers three services. The matching service enables a user to upload two ontologies and it computes matches which can then be revised. Detailed information in text or a graph visualization is also provided. The validator service enables the user to revise existing alignment uploaded in the Alignment format. Both services also support manual addition of new links. The API service enables to match two given ontologies from the programming language. While it also supports human validation of alignments, in comparison with Alignment the YAM++ Online does not provide collaborative mode for validation or link creation purposes. Additionally, it offers limited options to edit the Matcher configuration as it is only allowed to change the label properties, against which the Matcher calculates the similarities.

The next online ontology matching tool is LogMap [9]. LogMap is an ontology matching tool focused on scalability and reasoning. It implements algorithms for unsatisfiability detection and repair. Its online version is available as an online form. It enables one to enter two ontologies and specify whether or not to use mapping repair. If there are some unreliable mappings, a user can revise them

while a text information about each entity involved in mapping is available. In comparison with Alignment, LogMap lacks a collaborative mode for an alignment validation but on the other hand, it focuses on large ontologies and it aims at coherent alignments.

Authors of [10] introduced a collaborative ontology mapping environment Map-On. The Map-On tool supports creating of mappings between a database schema and existing domain ontology. The tool enhances mappings creation by different graphical layouts. First, it suggests mappings based on the database and domain ontology on input. Second, users can revise suggested mappings via a graphical ontology mapping interface. Similarly to Alignment, it provides a collaborative mode for mapping. On the contrary, Map-On is focused on mapping between a database schema and domain ontology.

Silk [11] is primarily focused on linked data integration, however, it can also support matching of ontologies. The Silk Workbench is a web application providing an interactive process of matching. A user can specify detailed linkage rules using declarative Silk-LSL. Silk targets at high scalability and performance. Silk is integrated into the Alignment platform as described in Section 3.3. In comparison, Alignment focuses more on visualization, collaboration and interactivity within its environment. Through the Silk Workbench, the user can validate entity matchings produced by the configuration of Silk-LSL but the user is not able to manually create links between the source and target dataset.

The Alignment server enables a user to access the Alignment API [7] via a browser interface but one can also use other web service interfaces such as REST or SOAP. The server supports the user to maintain existing alignments (by rendering, evaluating, comparing etc.) and to match two ontologies using basic matching techniques. The user can also work within the mode of an ontology network which is a collection of ontologies and their alignments. In comparison with the Alignment platform, the Alignment server is more oriented to alignment management than to user interaction and collaboration.

Similar to the Alignment server Visual Ontology Alignment environment (VOAR), ref. [12] enables a user to maintain alignments. The difference is that VOAR focuses on a visualization aspect and it supports working with multiple alignments at the same time. There are three main functionalities. The alignment visualization provides different visualization modes such as intended trees, graphs etc. The alignment manipulation provides an extensive list of alignment operations similar to the Alignment server such as trimming, union, inverting etc. Unlike the Alignment server, it further supports the mapping edition where the user can add, suppress or edit mappings. Finally, users can also evaluate multiple alignments against the provided reference alignment. The results are visualized in a tabular view. While VOAR also focuses on a visualization aspect of ontology matching, Alignment further targets the collaborative aspect of ontology validation.

A novel method of alignment visualization is provided in the Alignment Cubes tool [13] which enables a user to interactively explore multiple ontology alignments at the same time. It aims at supporting ontology alignment evaluation where traditional computed measures such as precision and recall are not enough or where a reference alignment is missing. The interface is a 3D cube view with two modes. The similarity mode provides an overview of existing mappings and their similarity values and the mappings mode provides an overview of numbers of existing mappings for an inspected pair of entities. It thus supports a comparison of different matchers and an identification of alignment parts with many or few mappings. While this approach is more suitable for ontology alignment evaluation, it could also support a collaborative ontology alignment which is a part of the Alignment platform. Therefore, as future work, the Alignment platform could include such visualization mode into its collaborative mode of mapping validation.

There are many other ontology matching tools [14] which are provided as stand-alone applications. However, they are out of the scope of the article since they mostly do not provide ontology matching in online, collaborative and visualization modes as the Alignment platform does.

3. Alignment Platform Presentation

3.1. System Architecture

The Alignment platform is a hybrid, interactive and collaborative ontology and entity matching service. A user-friendly GUI is offered in order to match two ontologies/vocabularies. The service can provide suggestions on the target entities, based on a default or user defined configuration of similarity measures and basic NLP procedures. Users can select one of the suggested links for each entity, or they can choose any other link to the target ontology, based on their domain knowledge. They can also customize the similarity variables that are used to compare the two ontologies which result in the suggested links, based on their use case and preferences.

The platform enables the collaborative creation and validation of linksets, as multiple users can work on the same project simultaneously and interactively. The users can provide a positive or negative vote or comment on a particular link between two entities, thus validating the linksets. A typical workflow is presented in Figure 1.

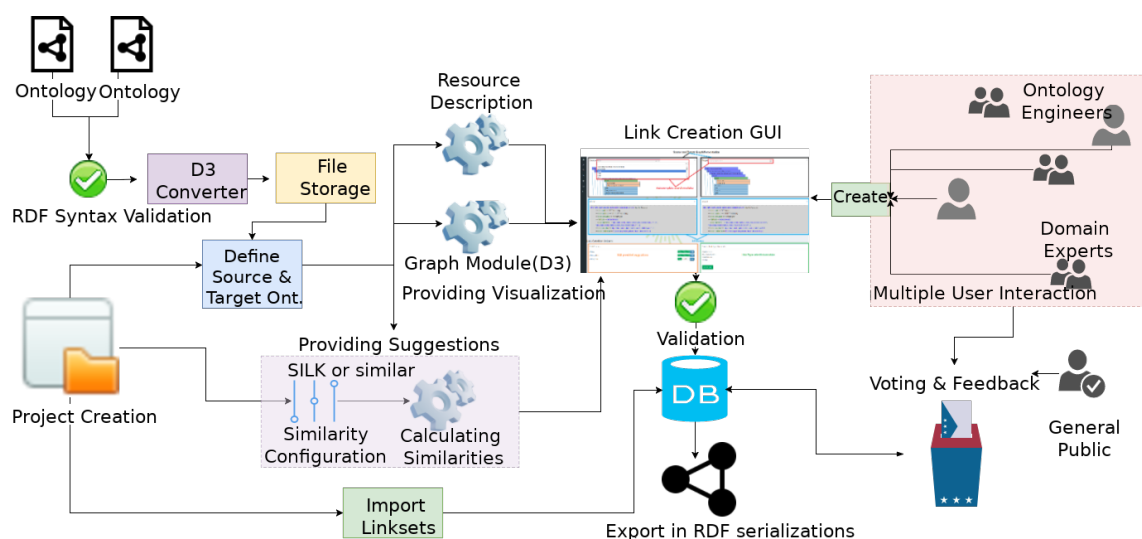


Figure 1. Alignment workflow.

The first step required is to upload the ontologies/vocabularies that the user wants in order to create a linkset (A linkset is a set of links or mappings between entities of two ontologies/vocabularies). The ontologies get validated automatically from the platform by using Quality check utilities, such as Skosify [15] and Rapper [16]. The validation step is required to ensure proper parsing and further processing of the uploaded ontologies on the platform. On successful validation of the ontologies, they are stored on the platform. The user may define the uploaded ontology as public, thus allowing other users to use the ontology in their own projects. Then the user can create a project in order to produce a linkset within the platform. This is followed by the user defining the source and target ontology from the selection list. Both user-uploaded and public ontologies are selectable through the list. The user may also define the similarity algorithm configurations that will be used for the system-provided suggestions. Currently only Silk Framework can be used as a suggestion provider, however, in the future version of the platform, the user will be able to choose alternatives such as YAM++ or Alignment API as suggestion providers as well as select a mixture of multiple providers. Furthermore, the user can provide public access to the project, thus allowing multiple users to work collaboratively. Upon creation of the project, the platform calculates similarities between the entities of the ontologies and renders the GUI. None of the suggestions provided by the system is realized as a valid link, unless some user decides to create a link from a predefined list of link types, or a user-defined link

type. Finally, produced linksets can be exported, or sent for crowd-sourced validation, through the Voting service.

3.2. GUI

Users should not be overwhelmed with too much information, but just enough in order to decide if a mapping should be created or not, as previous studies have shown [17]. We designed a minimal GUI with utilities to aid users, either domain or ontology engineering experts on the linking workflow as presented in Figure 2. On the “Create Links” page, both source and target ontologies are rendered as expandable/collapsible hierarchical tree graphs. Detailed element description and system-generated suggestions are provided as help for the linking process. Users can select a link type from predefined, grouped RDF links or a custom link type. An overview of the created links is also presented, where a user can edit or delete the produced links on the same page. A sample view of the GUI is shown in Figure 3.

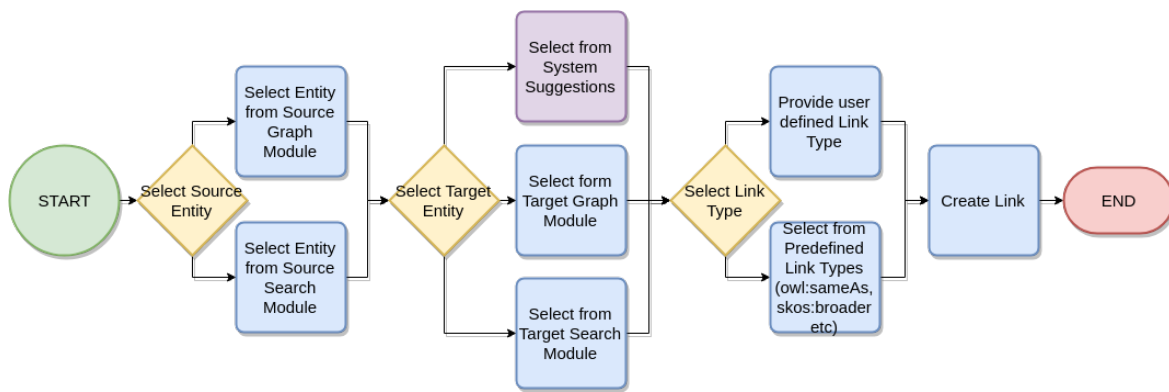


Figure 2. Link creation workflow.

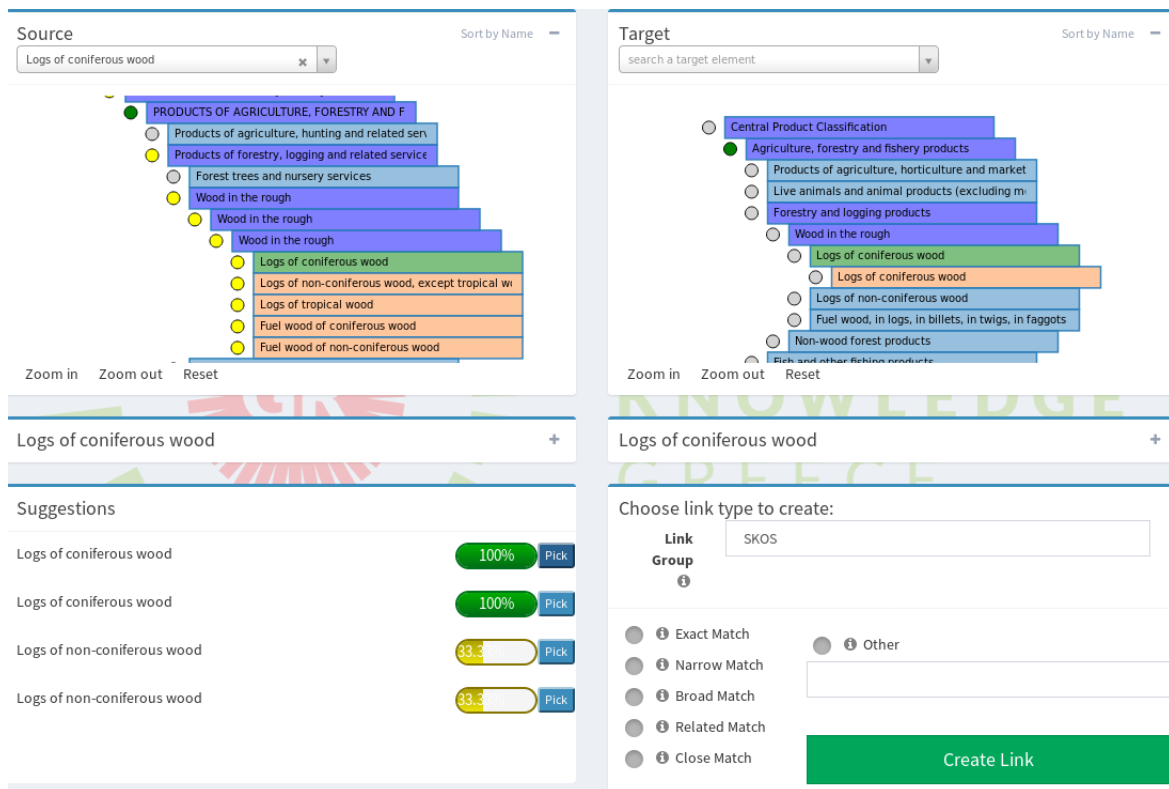


Figure 3. Alignment GUI.

3.2.1. Graph Module

By definition, the RDF data model represents a graph. Thus, ontologies and other RDF datasets can be represented as such. In the case of ontologies and vocabularies, hierarchical graphs can be formed, utilizing relationship properties as links between the elements of the ontology. The Graph Module of the Alignment Platform implements the visualizations of the ontologies by using the D3 JavaScript library (<https://github.com/d3/d3/wiki>). D3 can parse data in JSON format using a specific nested structure (A modified version of <https://bl.ocks.org/mbostock/1093025> is used to create the collapsible indented tree graph shown in Figure 4). We have developed a converter to transform RDF serializations or OWL ontology representations in that particular structure. The converter uses the SKOS semantic relationships properties (or OWL equivalent), such as `skos:broader` or `skos:narrower`, to realize the nested hierarchical structure of the graph.

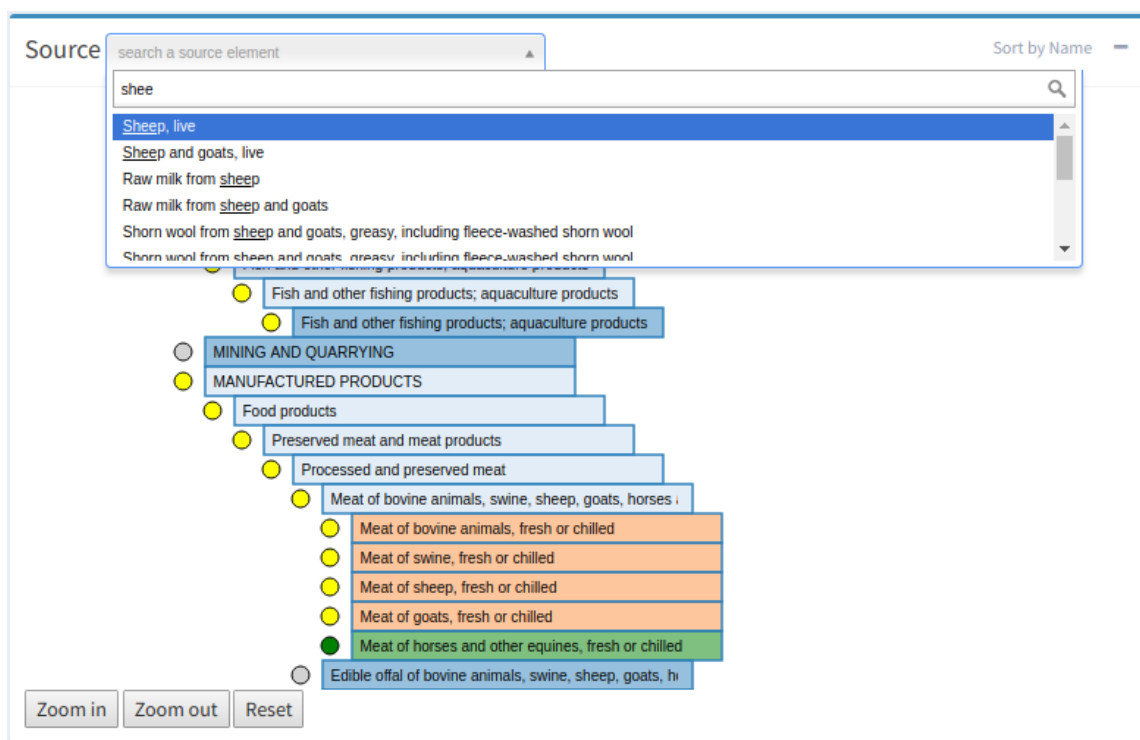


Figure 4. A close up of the Graph Module.

The converter is executed on the ontology importing step. The resulting JSON formatted file is cached to further increase performance of the renderer. Element coloring follows a predefined scheme, denoting different properties. For instance, blue colored elements contain child elements/nodes, while orange elements are terminals. The user can also control the zooming and panning of the graph by using the respective control buttons.

Above both of the graphs, there is a search bar, with an autocomplete function enabled. Through this bar, the user can search for a particular entity based on its label. On selection, the respective element is highlighted on the graph and a detailed description is presented.

On the left side of each element of the graph there is a circular indicator. This indicates the “linking” status of a particular element. The color gray indicates that there are no links for this element, either existing or suggested; the color yellow indicates elements for which the system has provided suggestions and green indicates elements that have been linked with an entity from the target graph. In the case of multiple users working on the same project, by considering the indicators the users can skip already linked elements, focus on “easy” to link elements or check what links have been created for a specific element. This functionality will be extended to indicate conflicts or validation errors.

3.2.2. Detailed Entity Information

The user can get the complete description of the selected entity. The description is displayed below the graph. A SPARQL DESCRIBE query is run on the selected element and the result is presented as a user-friendly and readable infobox as shown in Figure 5 or in raw RDF in multiple serializations. Users can click on resources describing or linked with this element to get additional information, therefore getting a comprehensive description of the selected entity.



Figure 5. Detailed entity information.

3.2.3. System Suggestions

Below the descriptions section, there is a panel with the similarity scores; these are calculated as described in Section 3.3. On the selection of a source element, the module retrieves the matched entities from the scores graph, if any, and displays them to the user. The scores are presented in score descending order, and a button on the right side of the score of each entity selects the related target entity on the graph, and presents its description in the target infobox section. If no suggested target entity is found or the user decides that none of the proposed entities are related to the selected source element, the user can utilize the Target Graph module to select a target entity. In Figure 6 we see a snapshot of the System Suggestions Module. The user has selected an entity from the source graph with the label “Noils of wool or of fine animal hair”. The system has found six candidate entities (five are shown here) from the target graph based on the similarity configuration of the project.

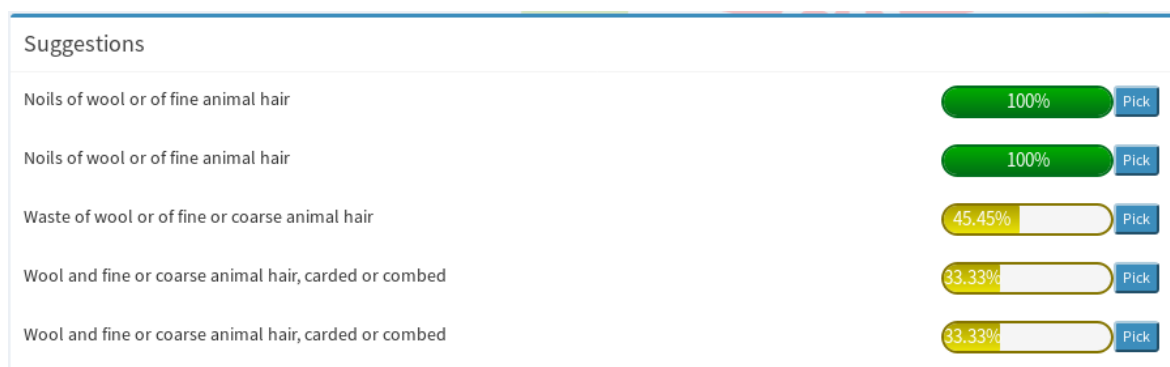


Figure 6. The system suggestions module.

3.2.4. Link Type Option

On the Semantic Web, different type of links express different semantics between entities. For example, two entities may represent the same concept so they could be linked via a `skos:exactMatch`, or `owl:sameAs` link, or concept A could be a broader match of concept B, so it could be connected via `skos:broadMatch`. The user should be able to choose the link type.

Through the Link Creator module, the user can select the link type to create between two entities, these can be selected from a short list of predefined link types, or a user may define his own link type

on the according text field. Link types are organized in discrete groups: SKOS, OWL and RDF related link types, in order to minimize confusion for the user. The Create Link button will create the new link, if the specified link passes the validation criteria. For instance, if the user tries to create an existing link triplet, the system will reject the link. The user is notified in either case through on-screen notifications. The module is presented in Figure 7.

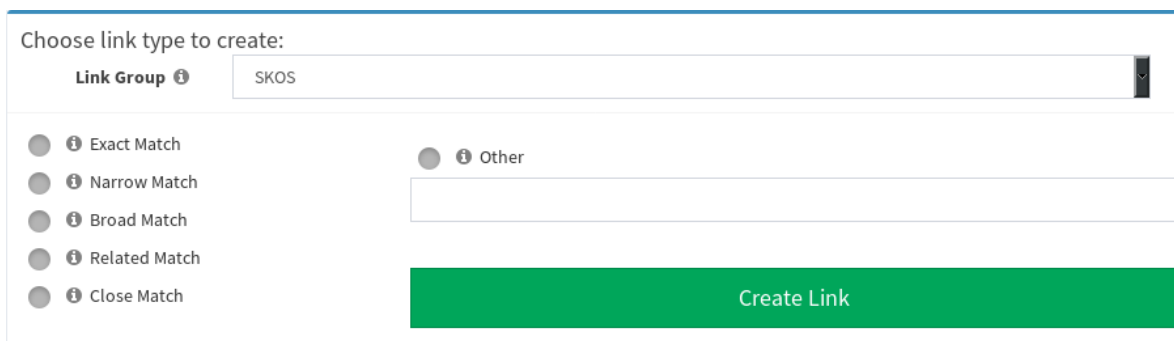


Figure 7. Link Type Selection and Link Creation Module. The user can provide a custom link type by selecting the “Other” option and define the link type on the respective field.

3.2.5. Created Links Monitor

On the last panel of the GUI, presented in Figure 8, the user can monitor the links created on the project, which are presented on a searchable and orderable table. The user can delete created links by clicking on the “X” button and export them into different RDF serializations. The user may also import links in various RDF serializations, that may have been produced by other automated or manual matching tools.

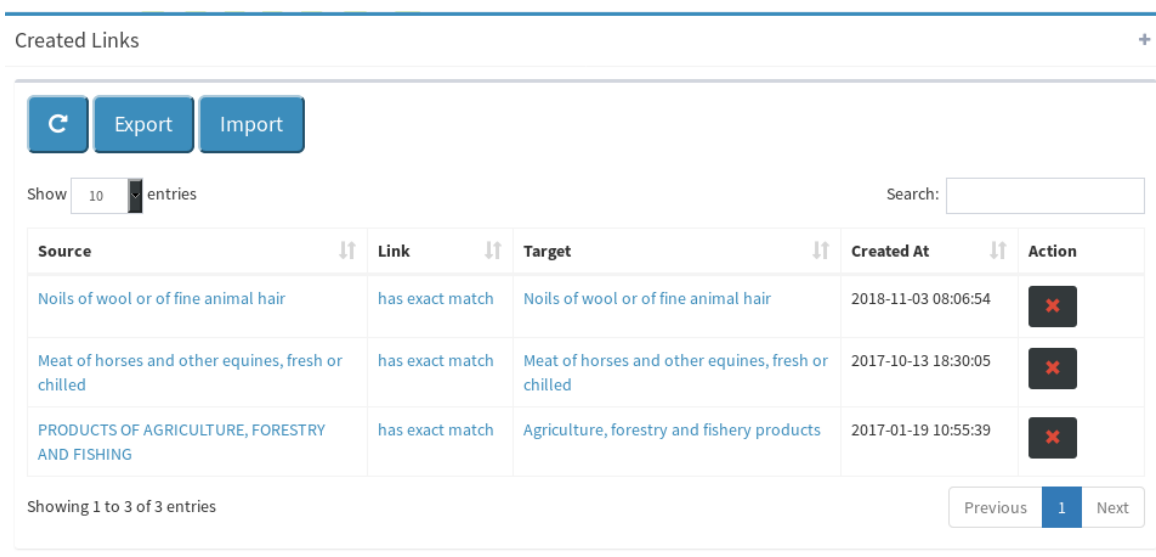


Figure 8. Created links monitor.

3.3. Calculating Similarities

Alignment integrates with the Silk Linking Framework [11], as a backend to calculate similarities between entities from different ontologies or vocabularies. The similarities found are used as suggestions for the manual linking part of the process. To do this, a Silk configuration file containing the comparison specifications in the Link Specification Language of Silk (Silk LSL) and XML form has to be created and executed by Silk. The user can either select the default application settings as the

comparison specifications, or can create user specified configuration file, customizing the similarity metrics to be used for the comparison or import a Silk LSL file.

The comparison process calculates similarities on predefined or user provided properties of the entities of the source and target ontologies. The process uses suitable similarity algorithms that are provided in Silk, like the Levenshtein distance metric, the Jaro-Winkler distance metric, the Dice coefficient metric or the soft-Jaccard similarity coefficient metric.

The comparison is exhaustive and for every entity of the source graph, all entities from the target graph are examined. This is the automatic part of the linking process and it has to be executed separately, before the user can manually select and link entities between the two graphs. The whole process is carried out as a background process and the user is notified upon completion. Silk is configured to store the calculated similarities using the Alignment format [18].

3.3.1. Default Configuration Parameters Choice and Validation

Finding the most suitable algorithm for accurately predicting similarities between entities is a complex task, which includes multiple string transformation steps and a variety of similarity algorithms to choose from. Calculating similarities scores may take considerable time to complete with unknown results, in a “trial and error” routine. At this point we should state that there is not a “one size fits all” formula.

We evaluated and validated settings to be used as default for Silk. For the purposes of the OpenBudgets.eu project, domain experts such as public officers and economists committed to provide and evaluate linksets between SKOSified codelists (<https://goo.gl/DoQBOQ>). A linkset is a set of RDF links. Code Lists are a key part of fiscal datasets. They are prescribed controlled vocabularies, that serve for the coding of concepts that can be expressed in many ways. In the scope of OpenBudgets.eu project, several EU or national level codelists were described using the SKOS Vocabulary, as RDF Data Cube dimension codelists as described in [2] and then linked to each other where possible. In some cases, links between codelists was provided by authoritative organizations such as the Eurostat’s RAMON mappings ([http://ec.europa.eu/eurostat/ramon/relations/index.cfm?TargetUrl=LST_LINK&StrNomRelCode=CPA%202008%20-%20CPC%202&StrLanguageCode=EN&StrOrder=2&CboSourceNomElt=&CboTargetNomElt=](http://ec.europa.eu/eurostat/ramon/rerelations/index.cfm?TargetUrl=LST_LINK&StrNomRelCode=CPA%202008%20-%20CPC%202&StrLanguageCode=EN&StrOrder=2&CboSourceNomElt=&CboTargetNomElt=)) of CPA (Classification of Products by Activity URL: <https://goo.gl/EknCdy>) to CPC (Central Product Classification URL:<https://goo.gl/Rx7qfg>). The linksets they produced are considered to be a golden standard in our approach, in order to evaluate and validate the similarity scores provided by Silk. In fact, domain experts are not expected to find or define similarity algorithms on their own. For this reason we tested different parameters of Silk configuration settings in order to find the most suitable and evaluate the automated part of the procedure, considering high recall percentage and the speed of the completion of the process.

Specifically, for N entities in CPA(5522) codelist and M in the CPC(4409) codelist, the total number of the possible links is obviously the cartesian product $P = N \times M$. RAMON mappings contain more than 3.8 K links between the two codelists, while the number of possible links is near 25 M. So, the links contained in the RAMON linkset are considered as validated positive, while the rest are considered as validated negative.

We produced several Silk Link Specification Language (Silk-LSL) files, based on four different similarity measures and different settings for each similarity measure. Dice Coefficient metric is a token-based text similarity metric. Given two sets of tokens X and Y , the Dice coefficient is defined as

$$sim_{Dice\ coeff.} = \frac{2 \cdot |X \cap Y|}{|X| + |Y|} \quad (1)$$

The Jaro [19] similarity metric is an edit-based string similarity metric. Given two strings s_1 and s_2 , the Jaro distance is defined as (https://en.wikipedia.org/wiki/Jaro-Winkler_distance):

$$sim_{Jaro} = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & \text{otherwise} \end{cases} \quad (2)$$

where

- $|s_i|$ is the length of string s_i ;
- m is the number of matching characters;
- t is the number of transpositions

Two characters are considered as matching only if they are the same and in a position within the string not further than $\frac{\max(|s_1|, |s_2|)}{2} - 1$. The number of transpositions is defined as the number of matching, but in different sequence order characters divided by 2.

The Jaro-Winkler similarity [20] was also considered. It is a variation of the Jaro similarity metric. It uses a prefix scale p that promotes the ratings of strings that match from the beginning. Given two strings s_1 and s_2 , their Jaro-Winkler similarity is defined as:

$$sim_{Jaro-Winkler} = sim_{Jaro} + [lp(1 - sim_{Jaro})] \quad (3)$$

where

- sim_{Jaro} is the Jaro similarity of the two strings;
- l is the length of the common prefix at the beginning of the string (maximum four characters)
- p is a constant scaling factor

Finally, the soft Jaccard Coefficient is a hybrid string similarity metric. It combines the Jaccard similarity index and the Levenstein similarity metric. Given two strings with a set of tokens X and Y , the Jaccard similarity metric is defined as:

$$sim_{Jaccard} = \frac{|X \cap Y|}{|X \cup Y|} \quad (4)$$

The soft Jaccard similarity metric allows tokens within a Levenstein distance to be considered as the equal.

We ran the Silk Single Machine version 2.7.1 using Silk-LSL formatted settings files against CPA and CPC as source and target codelists respectively in order to reproduce RAMON linkset in an automated way. Let TP (True Positive) be the number of the positive links retrieved by Silk and are contained within the RAMON linkset, then the rest are the FP (false positive). In addition, let FN be the set of links contained by the RAMON linkset but not retrieved by Silk. We consider *recall* (RE) as the ratio of TP links over the sum of total number of TP and FN links.

$$RE\% = \frac{TP}{TP + FN} \cdot 100\% \quad (5)$$

Precision (PR) metric is defined as the ratio of TP links over the sum of the TP and FP links.

$$PR\% = \frac{TP}{TP + FP} \cdot 100\% \quad (6)$$

Precision and recall metrics, calculated for each configuration file are shown in Table 1.

Results are better explained in Figure 9 where the 6th configuration achieves the best score, with 63.78% Recall score and time for completion of 73 s, which is a reasonable time considering the size of both graphs (near 5000 entities each).

By combining all methods, a small improvement in recall measure is reported, with a disanalogous cost in computation time. Most of the *TP* retrieved links between all methods overlap. We examined this case by calculating the mutual and unique positive links retrieved by all methods. The results are better explained in Figure 10.

Checked set in red is the validated linkset from RAMON, while the rest are the positive linksets retrieved by Silk using Silk-LSL configuration files as denoted by the name of each set. The first component represents the similarity algorithm used and the second the threshold set.

Table 1. Values marked with red and green, are the worst and best measure in each metric respectively.

	Similarity Measure	Threshold	Max Distance	PR%	RE%	Time (s)
1	Dice Coefficient	0.0	null	0.65	9.35	700
2	Dice Coefficient	0.1	null	39.59	23.45	14
3	Dice Coefficient	0.2	null	26.33	33.34	28
4	Dice Coefficient	0.3	null	21.13	41.96	27
5	Dice Coefficient	0.4	null	13.32	54.74	40
6	Dice Coefficient	0.5	null	8.77	63.78	73
7	Jaro Distance	0.0	null	43.54	16.18	69
8	Jaro Distance	0.1	null	34.92	21.71	69
9	Jaro Distance	0.2	null	14.02	32.85	65
10	Jaro Distance	0.3	null	3.46	44.25	68
11	Jaro-Winkler Distance	0.0	null	40.83	17.63	536
12	Jaro-Winkler Distance	0.1	null	21.56	28.85	475
13	Jaro-Winkler Distance	0.2	null	6.59	44.12	496
14	Jaro-Winkler Distance	0.3	null	3.66	49.65	569
15	Soft Jaccard Coefficient	0.0	2	39.84	21.32	18
16	Soft Jaccard Coefficient	0.1	2	39.64	21.92	20
17	Soft Jaccard Coefficient	0.2	2	36.45	26.43	52
18	Soft Jaccard Coefficient	0.3	2	30.79	31.68	97
19	Soft Jaccard Coefficient	0.4	2	19.78	40.28	140
20	Soft Jaccard Coefficient	0.5	1	14.49	50.61	184
21	Soft Jaccard Coefficient	0.5	2	11.47	52.12	202
22	Soft Jaccard Coefficient	0.5	3	4.93	52.69	176
23	Soft Jaccard Coefficient	0.5	4	2.88	40.17	237
24	Soft Jaccard Coefficient	0.5	5	1.93	27.47	196
25	All combined			1.11	75.31	NA

A big fraction of the validated links was not retrieve, however, use case specific modifications on Silk configuration can improve these results. In our case by including “service” and “product” at the stop-words list showed a 15% improvement in recall. Nevertheless, our goal was to try to find a generic approach, covering most use cases and as a result these improvements were excluded from the final default configuration. Advanced users are able to fine tune Silk Configuration as described in Section 3.3.2.

Because we only used string similarity algorithms in our approach, it was expected that precision metric would be affected negatively. Entities in codelists sometimes have the same labels, but they differ in hierarchy level (This actually violates SKOS integrity constraints but it is a fact for a number of Concept Schemes). String matching as described above ignores hierarchical structure. In this case, FP links increased with negative results in the precision score of the algorithm. The aim of the algorithm presented is to be able to provide at least one target entity as a suggestion for every source entity.

In the Silk Framework, a threshold of 0.0 means that only links that achieve a score of 100% similarity, based on rules specified, will be accepted. Increasing this value lowers the barrier of acceptance. We observed that for every similarity algorithm, by increasing the threshold, recall is also increased, however, precision is reduced. The results are shown in Figure 11.

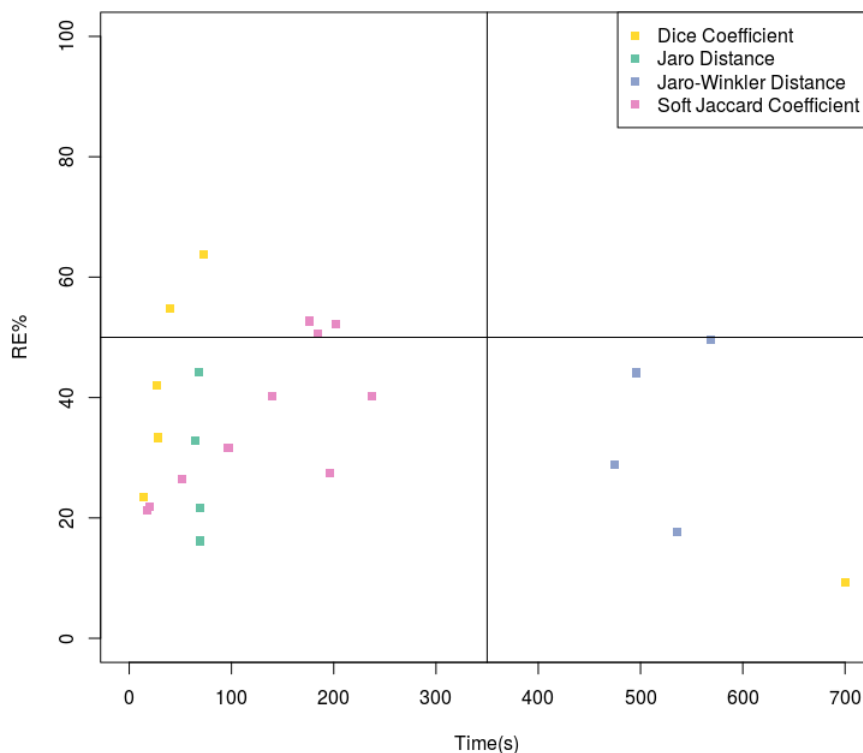


Figure 9. Recall over time. Top left corner shows the best performance.

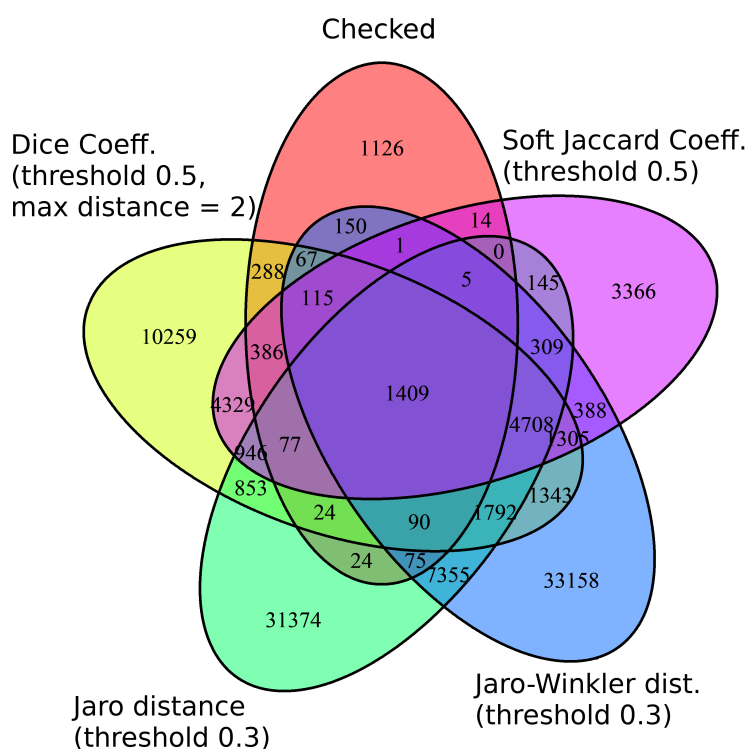


Figure 10. Venn diagram of top four methods, regarding Recall Score.

3.3.2. User Configuration

Besides the default comparison settings, a user can also modify the automatic part of the comparison process, by creating or importing a configuration file in Silk LSL format, for an Alignment

project between two graphs. These specifications are related to the comparison metrics, the linkage rule and the output of the comparison. In addition, these user-customized settings are automatically saved to the user profile or shared in public, so they can also be used on other projects.

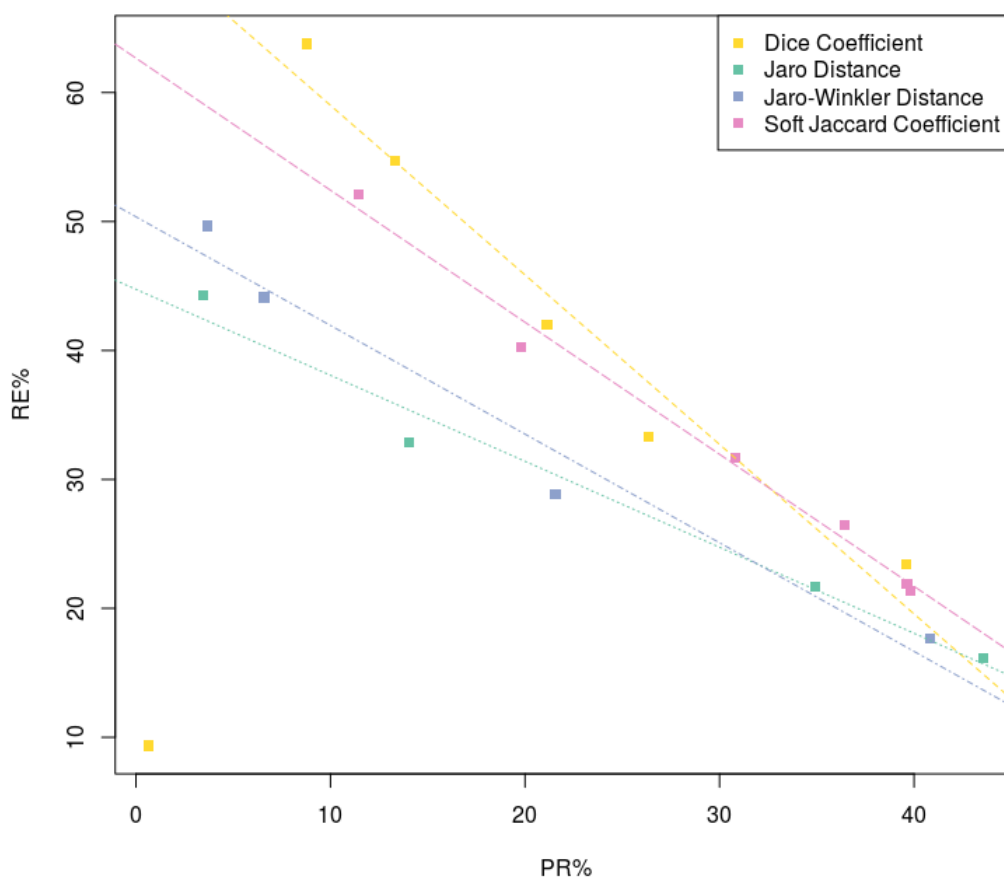


Figure 11. Recall over precision. In all configurations, increasing linking threshold drives to higher recall, reducing precision.

3.4. Integration, Collaboration and Social Features

User login is performed by using a variety of OAUTH capable services such as Github, Google or Facebook services, or by using the built-in registration system. Multiple users can work on the same project, enabling faster completion of projects. Additionally, linksets can be exposed to open voting to get validation.

3.4.1. Integration with Other Services

Created links can be exported as single files in different RDF serializations with the platform offering two additional options to provide linksets in different applications. Public linksets are available via a SPARQL Endpoint for Semantic Web applications and a REST API, as an access point for common applications.

3.4.2. SPARQL Endpoint

In order to be able to integrate with Semantic Web applications, a SPARQL Endpoint is implemented using the ARC2 PHP Library (<https://github.com/semsol/arc2>). With this approach we can reuse the existing MySQL database, without having to replicate the database in an RDF triple store, even though at the present time the endpoint supports only a subset of the SPARQL(1.0/1.1) features.

3.4.3. API

By design, the platform should support integration with legacy applications using APIs, in a Semantic Web agnostic manner. In order to comply, a REST API interface was developed. The API accepts HTTP post requests on the URL [http://\[ip:port\]/api/{function}](http://[ip:port]/api/{function}) where function can be string, IRI, ontology or ontologyIRI. The first two functions can be used to look up a specific entity or a set of entities. For instance, we can provide the label of an entity or its IRI, and get all the matched links that the provided entity has on all or specific ontologies. The other two functions are used to get information about a specific ontology that exists on the platform or to acquire all the matched links against a target ontology. The user can select from various response formats, like CSV or JSON.

3.4.4. Working on the Same Project

There are projects that may contain a vast number of entities which need to be linked, making them a bad candidate for manual linking. However, the effort expended is reduced by employing multiple users [21]. One of the requirements of the platform is to have the ability to host multiple users per project allowing these users to act simultaneously and interactively so as to develop linksets between ontologies quickly. The projects can be private or public. The public projects are visible and accessible to all users, so they can work collaboratively. Entities with existing links are marked through indicators to prevent the duplication of work and errors. Links are attached to the creator to enable filtering and monitoring of work completed or to prevent vandalism [22].

3.4.5. Crowdsourcing Link Validation

For validation purposes, the created links can be exposed on to public voting; this means that users can vote positively or negatively for a link and may also comment on each link, providing feedback for other users. A voter can request a subset of the links, in the form of a poll, in order to start the voting process and each subset is different from the other as it is created upon request of the voter. The pool of links is firstly reduced by the links the voter may have already voted for and then picked randomly from the pool. The size of the chunk can be defined by the project owner, affecting the voting session duration and the overall user experience.

After creating the voter specific poll, the links composing the subset are presented in succession to the voter. In fact, the interface is designed to be user-friendly and hide the underlined RDF data modeling. Each entity and link type is presented by its label in the voters language if it is available. As a result, the particular RDF triple is represented as a sentence as shown in Figure 12.

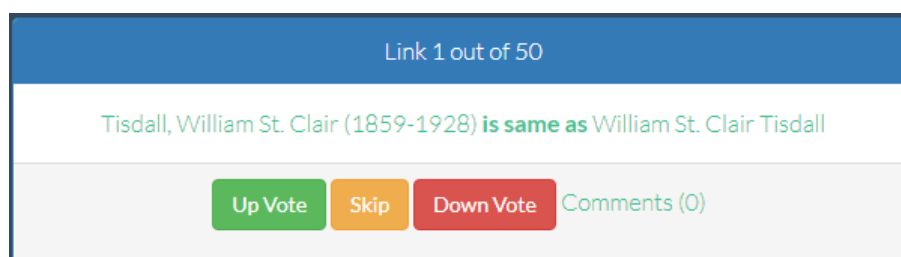


Figure 12. Voting interface.

Each part is clickable pointing to the resource IRI, enabling the user to retrieve more information. This requires the IRIs to be dereferenceable. To prevent biasing the voter, the exact current score of each link is not shown. The user can upvote, downvote or skip a link and in the case of downvoting a link, user feedback is requested, to classify the reason for downvoting. Feedback can be classified as (a) Unrelated, to mark false positive links, (b) Semantic error, to indicate a wrong link type (e.g., it should be `skos:narrowMatch` and not `skos:exactMatch`) and (c) Other, to cover the remaining cases, where the user can provide feedback about the reason for rejection. After the user has completed

the poll, an overview is shown which allows the user to change a vote, provide feedback, or revisit skipped links.

4. Evaluation

4.1. Link Creation Module Evaluation

As described in Section 3.2 the Link Creation module was designed to cover the needs of both semantic web experts, or domain experts with no previous knowledge of ontology engineering. So we conducted two experiments, testing the user experience and the outcome of the linking project. The first was targeted to Junior Level Ontology Engineers (graduate students) and the second targeted to public officers, experts in the fiscal data domain. Both were supervised by experienced Senior Ontology Engineers.

4.1.1. Junior Level Ontology Engineers

The scope of the experiment was to provide mappings between two codelists, from the Administrative Classification of the Municipality of Athens (hereafter Codelist A instances) and the Administrative Classification of Greek Municipalities (hereafter Codelist B) instances. (Codelist A: <https://goo.gl/UYrfPx>), Codelist B: <https://goo.gl/oxjLn>, Expert Mapping: <https://goo.gl/yiCTUw>. These codelists were used as dimension values in datasets described using the OpenBudgets.eu data model [23] an RDF DataCube [24] based fiscal data model. Thus, the mapping would enable direct comparison of fiscal data across Greek Municipalities [3].

The group consisted of 21 graduate students. The test was separated into two parts with an overall duration of two hours. The aim was to compare the two by time proposed methods to conduct links between SKOSified codelists. One was the use of advanced collaborative ontology editors, facilitating a platform like VoCol [25], and the other was the use of a GUI with system provided suggestions leading to the development of Alignment. In order to overcome the issue of the additional time required to get used to the VoCol platform, we decided to simulate the VoCol user experience, leaving out the collaborative part. To simulate the VoCol user experience, participants had the two codelists open with a text editor and created a new file to commit the linksets. Both codelists were given in Turtle format for easier manipulation and commitments to the linkset file were requested to be serialized in Turtle. So for each instance of the source Codelist A, the participants had to search for a compliant instance on the target Codelist B, and then commit the link on the linkset file on the following format:

```
<c1A_instance_IRI> skos:broadMatch <c1B_instance_IRI>.
```

Participants were limited to a time of one hour to conclude their linking projects, validate, fix and finally send their linksets. The validation part was made by using the Rapper utility [16], as this is the case for VoCol. Errors were fixed by the participants before sending the final linksets.

Afterwards, the participants were asked to complete the same task by using the Alignment platform. They created their own projects and then they had the same time limit to conclude their work. Each user worked separately, so we did not test the collaborative features of the platform. Bugs found on the platform and notices were tracked by assistants to give feedback to the developers team.

Out of the 21 submitted linksets produced with the text editor, 13 were unable to be parsed due to different kind of syntax errors. Some files contained invalid characters, other were missing the "<>" brackets, or the "." Turtle specific triple ending. Additionally, five of the linksets used concepts of the target codelist as subjects and vice versa, thus inducing semantic errors. Similarly, the Rapper utility provides syntax validation but the user has to understand and fix the error manually. In contrast, by using the Alignment platform, syntax errors are eliminated as the serialization of the links is handled and validated by the platform. Additionally, semantic errors were not present, as the platform always sets as subject resource an instance of the source ontology and vice versa. The utilities that Alignment offers, helped the participants to create linksets in a more efficient way. In fact, the majority of the participants (76.19%) reported that they enjoyed the session with Alignment. Likewise, 90.47%

answered that they would use the platform for linking projects again, but at the same time 61.9% of the participants stated that there was still room for improvement. Consequently, we plan to conduct a more analytical evaluation of this trial, in order to evaluate overall improvement of user scores in terms of precision and recall, using the produced linksets by the students and the complete mapping provided by an Ontology Engineering Expert. The overall conclusion of the trial is that while VoCol is an excellent platform to maintain and collaboratively create an ontology, when it comes to Ontology Alignment and especially employing domain experts with no prior knowledge of the RDF concept, it seems inadequate.

4.1.2. Domain Experts

The scope of the trial was to produce a linkset between the EU categorization system for funded projects with those of individual EU countries, in order to enable straightforward fiscal analyses. This was realized by building a linkset between the Czech codelist (44 items) to that of Europe (142 items). In order to ensure the quality of the linkset, we involved two domain experts, working separately, using the Alignment platform. They followed detailed guidelines which also included a short manual on how to use the Alignment GUI and the instruction that experts should prefer certain types of links, i.e., there was the following preference `skos:exactMatch`, then `skos:narrowMatch`, `skos:broadMatch` and then the rest.

Both experts interlinked 32 of the same items while expert one linked 84% (37) items from the source codelist and expert two linked 82% (36) items from source codelist. While expert one employed all `skos` link types (out of all the 53 links) more or less uniformly (21 `narrowMatch`, 11 `closeMatch`, 9 `exactMatch`, 8 `relatedMatch`, 4 `broadMatch`), expert two created mainly `narrowMatch` links (116), plus eight `exactMatch` and one `broadMatch`, out of all the 125 links. Both experts managed 32 times to link the same two entities in one link and, more importantly, they managed to create the very same link 23 times with seven `exactMatch`, one `broadMatch` and 15 `narrowMatch`. The resulted linkset of 23 links represents the nucleus of the reference linkset (Czech Codelist <https://goo.gl/pKZpVR>, European codelist <https://goo.gl/9hCPZq>, Guidelines: <https://goo.gl/vRYc5r>, Results: <https://goo.gl/BEmzfb>). Since there were so many links created by only one expert (57% in the case of expert one and 82% in the case of expert two), we plan to let experts discuss those links which were not agreed on as so to extend the current reference linkset.

4.2. Link Validation Module Evaluation

To test the Voting module of the Alignment platform we conducted a workshop, engaging domain experts to validate a linkset between Wikidata and the National Library of Greece Authority Records, produced by an automated procedure.

Trial Set Up

Twenty-two participants, varying from young undergraduate librarians to experienced public Library Cataloguers, worked on the same project to validate a linkset produced by an automated procedure. The sum of 11K links was created by using string similarity algorithms, provided by Silk. However, in order to achieve high recall scores, the threshold was set to be low, this meant that configuration also introduced false positive links, and as a consequence, user interaction was needed. The linkset was imported on the platform within a public project created by the instructor, and after a brief presentation of the concept and a demonstration of the basic Voting System utilities, the participants started to validate the links using the platform. The project was configured to offer chunks with a size of 25 links each time a user requested a new Poll. Each participant had to complete at least one Poll. Assistants were logging bugs of the system and feedback from the participants about possible design pitfalls and 'nice-to-have' features.

After almost 20 min, a set of 395 unique links (3.5% of the total linkset) were validated using the Alignment Platform. An overview of the voting process is presented in Figure 13. The vast majority

of the participants completed just one poll while some more enthusiastic ones started more polls. The validation process helped us to improve our similarity algorithms configuration for the automated procedure. After the completion of the trial, participants had to complete a questionnaire based on the usability of the platform and the overall experience. About 73% of the participants said that they would like to include the platform on their workflow, while about 80% found that they felt it was easy to use the platform.

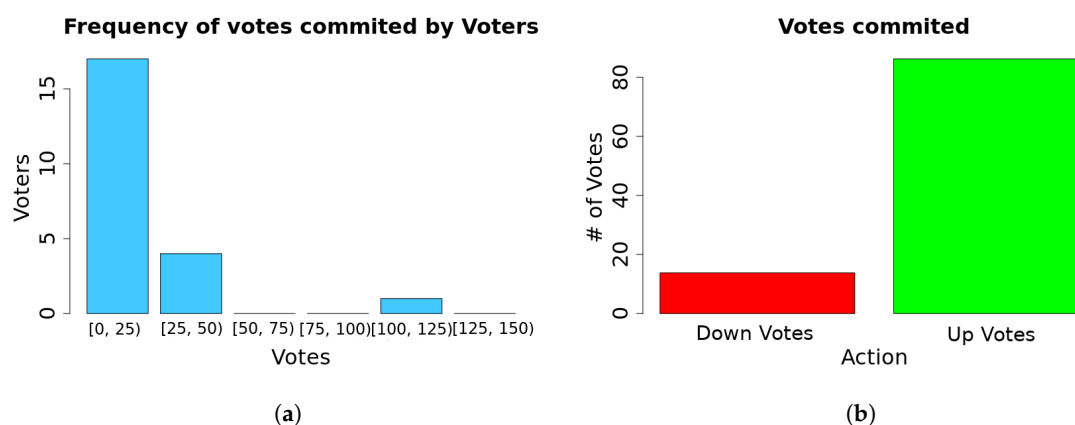


Figure 13. Voting Process Overview. (a) Frequency distribution of votes per user; (b) Domain experts found that 13% of the links presented to them were wrong.

4.3. Besides OpenBudgets.eu—Use Cases

The Alignment Platform has been used in several other cases apart from the OpenBudgets project. In the following paragraphs, the most notable examples are presented.

4.3.1. EveryPolitician Project

EveryPolitician (<http://docs.everypolitician.org/>) is a project that aims to provide data about every politician in the world. The project utilizes the community of Wikidata through the EveryPolitician WikiProject (https://www.wikidata.org/wiki/Wikidata:WikiProject_every_politician) and local events organized throughout the world. Within this context, a relevant event in the form of an editathon was organized in Greece by the EveryPolitician Project, the Open Knowledge Greece, the Greek Wikipedia User Group and Vouliwatch (<https://vouliwatch.gr/about/en>), an NGO in Greece that monitors the activity of the Hellenic Parliament. The goal of the event was to have complete and comprehensive data on the Hellenic Parliament politicians of the recent parliamentary term. In order to achieve this, a dataset is considered as complete and comprehensive if it contains information about the current position held (parliamentary term, district elected, party, district etc.) and biographical data (given name, date of birth, sex or gender, social accounts etc.). Vouliwatch offers an API with most of this information openly available. Instead of manually editing every record on Wikidata, we decided to import the dataset of Vouliwatch to Wikidata, meaning that a complete mapping between entities of both datasets was required. We exported the entities from Wikidata and described the Vouliwatch dataset in RDF format using FOAF. Both datasets were imported to the Alignment Platform. Participants created and validated collaboratively the mappings of the entities through the Alignment Platform. Finally, we developed a PyWikibot consuming both datasets and the mappings from the Alignment API, to import data on Wikidata automatically. At the end of the procedure, the completion rate of the Wikidata dataset reached 100% in most of the indicators (https://www.wikidata.org/wiki/Wikidata:WikiProject_every_politician/Greece/Parliament/completeness).

4.3.2. PhD Hub

The European PhD Hub (<http://phdhub.eu>) is an online platform that will provide PhD students, Researchers and Universities with opportunities for cooperation and funding from the public and private sectors, both in direct research funding and commissioned research from enterprises. The project consists of three specific outputs aimed at: (a) Establishing a quality framework for university-business cooperation, (b) Building an online infrastructure to increase research opportunities and their transferability and (c) Applying and mainstreaming the European PhD Hub. An advanced matchmaking procedure between the users of the platform and the research opportunities has been identified as a critical point for the success of the platform [26]. The matchmaking procedure is based on the conceptual linking of the research areas. These are represented as a Knowledge Graph, consisting of already established Scientific Classifications such as the MeSH [27], the ACM CCS [28] and STW Thesaurus for Economics [29] among others. Overlapping or related concepts were linked manually, using the Alignment Platform. Knowledge Management on the PhD Hub platform is handled by VocBench. Although VocBench offers a link validation GUI, it cannot be used to create links manually in a user-friendly environment as discussed in Section 2. The Alignment Platform can integrate with the backend of VocBench, SemanticTurkey(ST). Accordingly, it can import graphs from ST and export links back to ST after the validation process, extending the functionality of VocBench.

5. Summary

As described in Section 1, there are cases where automated ontology matching tools fail to produce either complete or partial matchings between ontologies. Thus, there is a need to produce linksets manually. In the case of confined size of ontologies, the process might be trivial for an experienced Ontology Engineer with excessive domain knowledge within the context of the ontologies in question. However, there are cases where the size of the ontology is non-negligible or there is the need to engage domain experts, which usually have no previous experience on the ontology matching task. As we have shown in Section 2, there is a gap in collaborative ontology matching tools and Alignment was developed to bridge this gap. We presented the Alignment platform, a novel service towards collaborative interactive ontology and entity matching on the Semantic Web World. This tool was developed based on the experience gained working on Horizon 2020 Openbudgets.eu project, in order to enhance domain experts of fiscal datasets, manually creating and evaluating links between heterogeneous ontologies. The service can be used as well to crowdsource linkset creation and evaluation. During our evaluation tests and use cases presented in Section 4, we demonstrated the Alignment Platform in various audiences, receiving overall positive reaction and comments for the user-friendliness of the application.

Furthermore, we are working on the development of a translation functionality to enable system aids for multilingual ontologies. In the next development cycle we will enable the capability to use alternative suggestion providers such as LIMES, YAM++, Alignment API, or others.

Author Contributions: Conceptualization, C.B. and I.A.; Software, S.K. and P.M.F.; Supervision, C.B.; Writing—original draft, S.K. and P.M.F.; Writing—review & editing, S.K. and O.Z.

Funding: This work has been supported by the OpenBudgets.eu Horizon 2020 project (Grant Agreement 645833).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SKOS	Simple Knowledge Organization System
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language

API	Application Programming Interface
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
OWL	Ontology Web Language
ACM CCS	Association for Computing Machinery Computing Classification System
MeSH	Medical Subject Headings
STW	Standard-Thesaurus Wirtschaft
GUI	Graphical User Interface
FOAF	Friend of A Friend Vocabulary
Silk LSL	Silk Link Specification Language
CPA	Classification of Products by Activity
CPC	Central Product Classification

References

1. Sabou, M.; Ekaputra, F.J.; Biffl, S. Semantic Web Technologies for Data Integration in Multi-Disciplinary Engineering. In *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*; Biffl, S., Lüder, A., Gerhard, D., Eds.; Springer International Publishing: Berlin, Germany, 2017; pp. 301–329.
2. Filippidis, P.M.; Karampatakis, S.; Koupidis, K.; Ioannidis, L.; Bratsas, C. The code lists case: Identifying and linking the key parts of fiscal datasets. In Proceedings of the 11th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), Thessaloniki, Greece, 20–21 October 2016; pp. 165–170.
3. Filippidis, P.M.; Karampatakis, S.; Ioannidis, L.; Mynarz, J.; Svátek, V.; Bratsas, C. Towards Budget Comparative Analysis: the need for Fiscal Codelists as Linked Data. In Proceedings of the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, 12–15 September 2016.
4. Shvaiko, P.; Euzenat, J. *Ontology Matching: State of the Art and Future Challenges*; IEEE Educational Activities Department: Piscataway, NJ, USA, 2013; Volume 25, pp. 158–176.
5. Karampatakis, S.; Bratsas, C.; Zamazal, O.; Filippidis, P.M.; Antoniou, I. Alignment: A Collaborative, System Aided, Interactive Ontology Matching Platform. In *Knowledge Engineering and Semantic Web*; Różewski, P., Lange, C., Eds.; Springer International Publishing: Berlin, Germany, 2017; pp. 323–333.
6. Stellato, A.; Turbati, A.; Fiorelli, M.; Lorenzetti, T.; Costetchi, E.; Laaboudi, C.; Van Gemert, W.; Keizer, J. Towards VocBench 3: Pushing collaborative development of thesauri and ontologies further beyond. In Proceedings of the 17th European Networked Knowledge Organization Systems Workshop (NKOS 2017), Thessaloniki, Greece, 21 September 2017.
7. David, J.; Euzenat, J.; Scharffe, F.; Trojahn dos Santos, C. The alignment API 4.0. *Semant. Web* **2011**, *2*, 3–10.
8. Ngo, D.; Bellahsene, Z. Overview of YAM++—(not) Yet Another Matcher for ontology alignment task. *Web Semant. Sci. Serv. Agents World Wide Web* **2016**, *41*, 30–49. [[CrossRef](#)]
9. Jiménez-Ruiz, E.; Grau, B.C.; Zhou, Y.; Horrocks, I. Large-scale Interactive Ontology Matching: Algorithms and Implementation. In Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012), Montpellier, France, 27–31 August 2012; pp. 444–449.
10. Sicilia, Á.; Nemirovski, G.; Nolle, A. Map-On: A web-based editor for visual ontology mapping. *Semant. Web* **2017**, *8*, 969–980. [[CrossRef](#)]
11. Volz, J.; Bizer, C.; Gaedke, M.; Kobilarov, G. Silk-a Link Discovery Framework for the Web of Data. In Proceedings of the 2nd Linked Data on the Web Workshop (LDOW 2009), Madrid, Spain, 20 April 2009.
12. Severo, B.; Trojahn, C.; Vieira, R. VOAR 3.0: A Configurable Environment for Manipulating Multiple Ontology Alignments. In Proceedings of the International Semantic Web Conference (Posters, Demos & Industry Tracks), Vienna, Austria, 21–25 October 2017.
13. Ivanova, V.; Bach, B.; Pietriga, E.; Lambrix, P. Alignment Cubes: Towards Interactive Visual Exploration and Evaluation of Multiple Ontology Alignments. In Proceedings of the International Semantic Web Conference, Vienna, Austria, 21–25 October 2017; pp. 400–417.
14. Euzenat, J.; Shvaiko, P. *Ontology Matching*; Springer: Berlin/Heidelberg, Germany, 2013.

15. Suominen, O.; Hyvönen, E. Improving the quality of SKOS vocabularies with Skosify. In Proceedings of the International Conference on Knowledge Engineering and Knowledge Management, Galway City, Ireland, 8–12 October 2012; pp. 383–397.
16. Beckett, D. The design and implementation of the Redland RDF application framework. *Comput. Netw.* **2002**, *39*, 577–588. [[CrossRef](#)]
17. Dragisic, Z.; Ivanova, V.; Lambrix, P.; Faria, D.; Jiménez-Ruiz, E.; Pesquita, C.; Groth, P.; Simperl, E.; Gray, A.; Sabou, M.; et al. User Validation in Ontology Alignment. In Proceedings of the 15th International Semantic Web Conference, the Semantic Web—ISWC 2016, Kobe, Japan, 17–21 October 2016; Proceedings, Part I; pp. 200–217.
18. Euzenat, J. An API for Ontology Alignment. The Semantic Web—ISWC 2004. In Proceedings of the Third International Semantic Web Conference, Hiroshima, Japan, 7–11 November 2004; McIlraith, S.A., Plexousakis, D., van Harmelen, F., Eds.; Springer: Berlin, Heidelberg, 2004; pp. 698–712.
19. Jaro, M.A. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *J. Am. Stat. Assoc.* **1989**, *84*, 414–420. [[CrossRef](#)]
20. Winkler, W.E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Available online: https://www.researchgate.net/publication/243772975_String_Comparator_Metrics_and_Enhanced_Decision_Rules_in_the_Fellegi-Sunter_Model_of_Record_Linkage (accessed on 15 November 2018).
21. Cordasco, G.; De Donato, R.; Malandrino, D.; Palmieri, G.; Petta, A.; Pirozzi, D.; Santangelo, G.; Scarano, V.; Serra, L.; Spagnuolo, C.; et al. Engaging Citizens with a Social Platform for Open Data. In Proceedings of the 18th Annual International Conference on Digital Government Research, Staten Island, NY, USA, 7–9 June 2017; pp. 242–249.
22. Geiger, R.S.; Ribes, D. The work of sustaining order in Wikipedia. In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, Savannah, GA, USA, 6–10 February 2010; pp. 117–126.
23. Mynarz, J.; Svátek, V.; Karampatakis, S.; Klímek, J.; Bratsas, C. Modeling fiscal data with the Data Cube Vocabulary. In Proceedings of the 12th International Conference on Semantic Systems (SEMANTICS 2016), Leipzig, Germany, 12–15 September 2016.
24. Cyganiak, R.; Reynolds, D.; Tennison, J. The RDF Data Cube Vocabulary. *W3C Recommendation (January 2014)*. Available online: <https://www.w3.org/TR/vocab-data-cube/> (accessed on 14 November 2018).
25. Halilaj, L.; Petersen, N.; Grangel-González, I.; Lange, C.; Auer, S.; Coskun, G.; Lohmann, S. Vocol: An integrated environment to support version-controlled vocabulary development. In *Knowledge Engineering and Knowledge Management, Proceedings of the 20th International Conference, EKAW 2016, Bologna, Italy, 19–23 November 2016*; Springer: Berlin, Germany, 2016; pp. 303–319.
26. Bratsas, C.; Filippidis, P.M.; Karampatakis, S.; Ioannidis, L. Developing a scientific knowledge graph through conceptual linking of academic classifications. In Proceedings of the 2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), Zaragoza, Spain, 6–7 September 2018.
27. Rogers, F.B. Medical subject headings. *Bull. Med. Libr. Assoc.* **1963**, *51*, 114–116. [[PubMed](#)]
28. Coulter, N.; French, J.; Glinert, E.; Horton, T.; Mead, N.; Rada, R.; Ralston, A.; Rodkin, C.; Rous, B.; Tucker, A.; et al. Computing classification system 1998: current status and future maintenance. Report of the CCS update committee. *Comput. Rev.* **1998**, *39*, 1–62.
29. Neubert, J. Bringing the “Thesaurus for Economics” on to the Web of Linked Data. In Proceedings of the 2nd Linked Data on the Web Workshop (LDOW 2009), Madrid, Spain, 20 April 2009.

