OWLS-SLR

An OWL-S Service Profile Matchmaker

Quick Use Guide (v0.1)

Intelligent Systems and Knowledge Processing Group
Aristotle University of Thessaloniki, Greece

Author: Georgios Meditskos, PhD Student

September, 2008

1 Preliminaries

(Note: The purpose of this guide is to present the basic configuration options of OWLS-SLR. More details about the background, the underlying matchmaking framework and the implemented algorithms have been submitted to a journal and are available upon request)

OWLS-SLR (SLR stands for Structural and Logic-based Reasoning) is a semantic Web service matchmaker written in Java that makes use of OWL-S Profiles. It follows the abstract Web service discovery approach, performing matchmaking based on inputs, outputs and non-functional properties. The matchmaking algorithm exploits not only the ontology subsumption hierarchy, but also the structural knowledge of the domain ontologies, such as sibling relationships among concepts.

OWLS-SLR is the prototype implementation of our semantic Web service discovery framework that adopts a conceptual model, following the WSMO Discovery Framework and using descriptions that are expressed as instances of the Profile concept of the Service Profile of the OWL-S 1.1 ontology. Some of the main features of OWLS-SLR include:

1. *Taxonomy-based matchmaking.* OWLS-SLR allows the existence of a Profile Taxonomy, that is, a subclass hierarchy of the OWL-S Profile concept. This is feasible by performing complete ABox DL reasoning on OWL-S advertisements and queries that are both represented as direct or indirect Profile ontology instances. OWLS-SLR has been defined on top of the Pellet DL reasoner [1].
2. *Structural ontological knowledge.* OWLS-SLR implements two concept distance measures, namely the *edge distance* and the *upwards cotopic* [2] distance. The former computes the distance of two ontology concepts based on the number of edges found on their shortest path. The latter determines the similarity of two ontology concepts taking into account the ratio of the common superclasses of the two concepts. In that way, we are able to match concepts not based only on logic-based DL reasoning, but also on the structural knowledge that the ontology provides, returning useful matchmaking results.

3. *Role-oriented matchmaking*. OWLS-SLR allows the annotation of the input and output parameters of the Profile concept with ontology roles. This is achieved by extending the Input and Output OWL-S concepts and exploiting the excellent classification capabilities of DL reasoning that is based on anonymous concept definition. The rationale behind this direction is to allow the ontology roles to play an important role during matchmaking, since they encapsulate important domain knowledge.
4. *Filtering.* The functionality of OWLS-SLR is characterized by a number of filters that allow users to control the matchmaking results according to the application domain and their preferences.

The matchmaking algorithm combines object-based structural matching techniques that are used in the domain of Structural Case-based Reasoning (SCBR) [3] with OWL Description Logic (DL) reasoning [4] over Profile instances. For each pair <query, advertisement> we define three levels of similarity:

1. *Taxonomical Similarity.* It is computed over the instance class membership sets of the query and the advertisement and denotes their similarity in terms of their taxonomical categorization in a Profile subclass hierarchy.
2. *Functional Similarity*. It is computed over the input and output sets of a query and an advertisement (signature matching).
3. *Non-Functional Similarity.* It is computed over the values of the common non-functional properties of a query and an advertisement.

The three similarities are aggregated into a single value, using the equation:

$$\overline{Sim} = \frac{a \cdot TS + b \cdot FS + c \cdot NFS}{a + b + c}, \qquad (1)$$

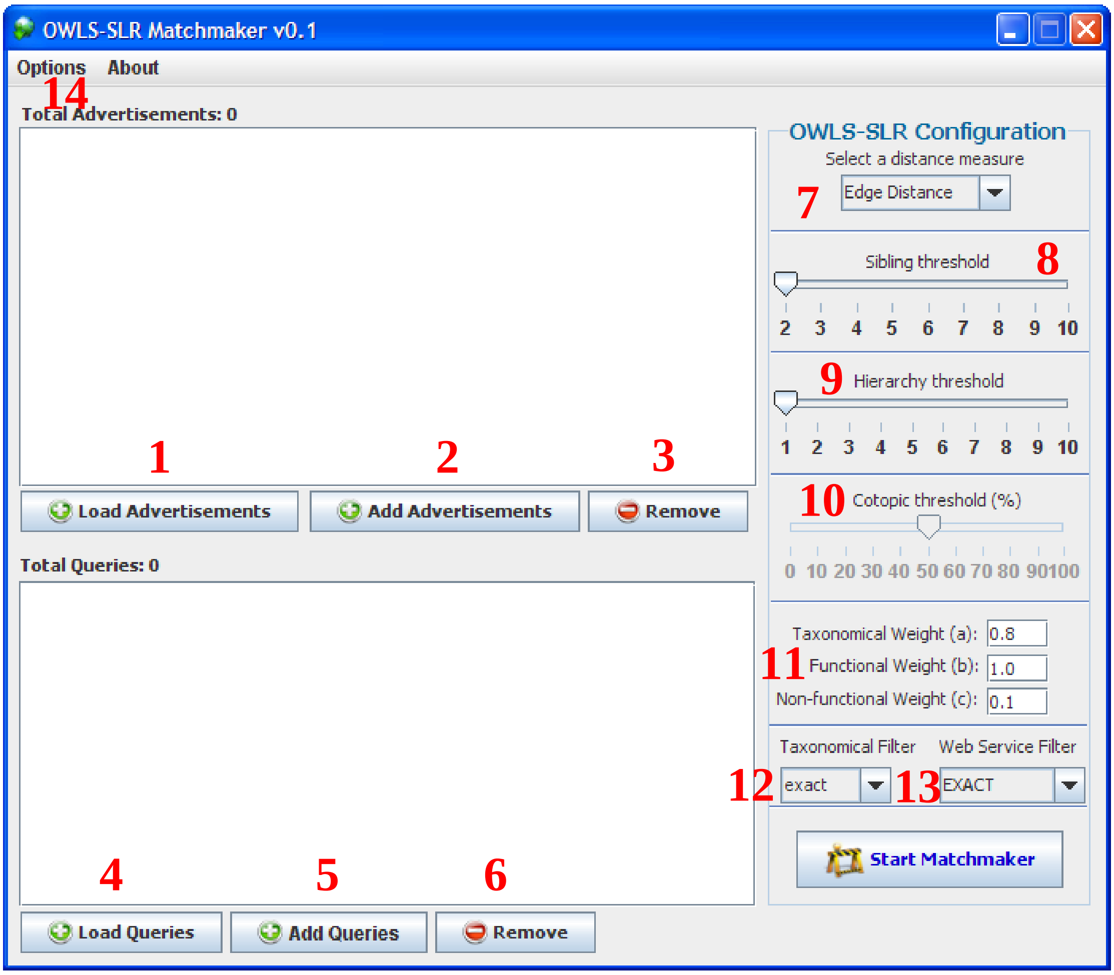
where *a*, *b* and *c* are normalized weights in [0..1].

The system was developed by Georgios Meditskos, PhD student in department of Informatics in Aristotle University of Thessaloniki, Greece, under the supervision of the assistant professor Nick Bassiliades.

OWLS-SLR is available for testing in [5], together with installation instructions and the datasets we used.

2 Main Window

Figure 1 depicts the main window of OWLS-SLR, where users can load OWL-S advertisements and queries, as well as to configure several options of OWLS-SLR.



Total Advertisements: 0

Total Queries: 0

Select a distance measure: Edge Distance

Taxonomical Weight (a): 0.8

Functional Weight (b): 1.0

Non-functional Weight (c): 0.1

Taxonomical Filter: exact

Web Service Filter: EXACT

Figure 1. The main window of OWLS-SLR.

More specifically:

1. **Load Advertisements**. Removes the already added advertisements from the list and loads a new set.
2. **Add Advertisements**. Adds one or more advertisements to the list.
3. **Remove**. Removes the selected advertisements from the list.
4. **Load Queries**. Removes the already added queries from the list and loads a new set.

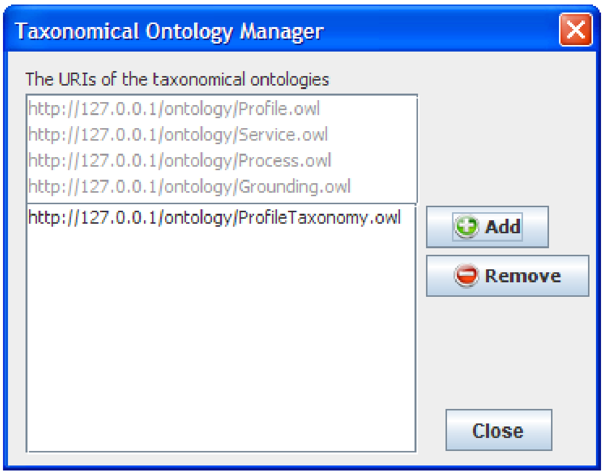
5. **Add Queries**. Adds one or more queries to the list.
6. **Remove**. Removes the selected queries from the list.
7. **Distance measure**. This is the distance measure that OWLS-SLR will use to compute the similarity among concepts during matchmaking. The two available options are the Edge Distance and the Upwards Cotopic.
8. **Sibling Threshold**. This option defines the maximum number of edges that we allow to exist among sibling concepts. For example, the edge distance between the direct subclasses of a concept is 2. This option is applicable only in the Edge Distance.
9. **Hierarchy Threshold**. This option defines the maximum number of edges that we allow to exist among concepts with a hierarchical relationship. This option is applicable only in the Edge Distance.
10. **Cotopic threshold**. The cotopic distance is defined based on the ratio of the common superclasses of two concepts to the total number of their superclasses. The cotopic threshold defines the minimum ratio. For example, a cotopic ratio equal to 50% means that two concepts should have in common at least half of their superclasses. This option is only applicable in the Upwards Cotopic distance.
11. **Weights**. Users may define the weights that are used in equation (1).
12. **Taxonomical filter**. This filter defines the hierarchical relationships that should exist among the taxonomical concepts of advertisements and queries.
13. **Web Service Filter**. This filter define the hierarchical relationships that should exist among the input and output concepts of advertisements and queries.
14. **Options Menu**. It is used in order to register the taxonomical ontologies of a collection (see section 2.2 and 2.3).

2.1 Loading the OWLS-TC_v2.2_revision_3 collection

The OWLS-TC_v2.2_revision_3 collection contains 1007 OWL-S advertisements and 29 queries. All the advertisements are defined as direct Profile instances (see [5] for dataset installation instructions). No special configuration is needed. The advertisements and queries can be loaded directly with the corresponding buttons, configuring OWLS-SLR appropriately.

2.2 Loading the OWLS-TC_v2.2_revision_3_(profile_taxonomy) collection

The advertisements and queries of the OWLS-TC_v2.2_revision_3_(profile-_taxonomy) [5] collection are defined with respect to a Profile taxonomy, namely the ProfileTaxonomy.owl ontology that exists in the ontology folder of the collection. The current implementation of OWLS-SLR requires this ontology to be explicitly defined in OWLS-SLR as a taxonomical. Otherwise, it will be considered as a domain ontology by the system, returning incorrect results. Users can register taxonomical ontologies through the **Options** → **Change Taxonomical Ontologies** menu at the top of the main window. Click the Add button and write the ontology URI, which is http://127.0.0.1/ontology/ProfileTaxonomy.owl. (Figure 2).



Taxonomical Ontology Manager

The URIs of the taxonomical ontologies

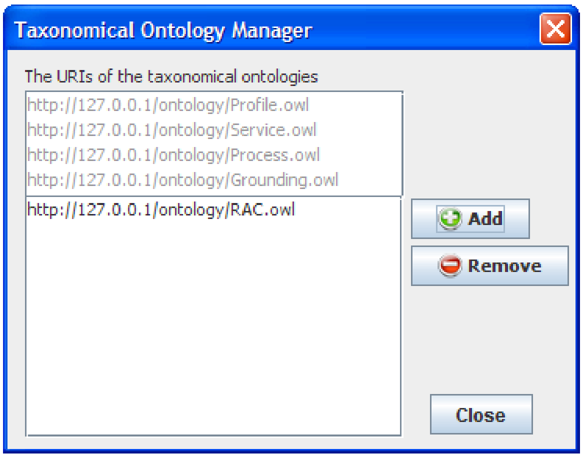
http://127.0.0.1/ontology/Profile.owl
http://127.0.0.1/ontology/Service.owl
http://127.0.0.1/ontology/Process.owl
http://127.0.0.1/ontology/Grounding.owl

http://127.0.0.1/ontology/ProfileTaxonomy.owl

Figure 2. Registering the ProfileTaxonomy.owl ontology.

2.3 Loading the RAC_dataset collection

In the RAC_dataset [5], the advertisements and queries are defined as direct Profile instances, and therefore, there is not a taxonomical ontology. However, the rac-based descriptions are based on an extension of the OWL-S Input and Output concepts of the Process ontology (see the RAC.owl ontology in the ontology folder of the collection). Therefore, the RAC.owl ontology should be registered as a taxonomical, since the Process.owl ontology is considered as a taxonomical. Users should follow the same procedure to the previous case, using the http://127.0.0.1/ontology/RAC.owl URI (Figure 3).



Taxonomical Ontology Manager

The URIs of the taxonomical ontologies

http://127.0.0.1/ontology/Profile.owl
http://127.0.0.1/ontology/Service.owl
http://127.0.0.1/ontology/Process.owl
http://127.0.0.1/ontology/Grounding.owl

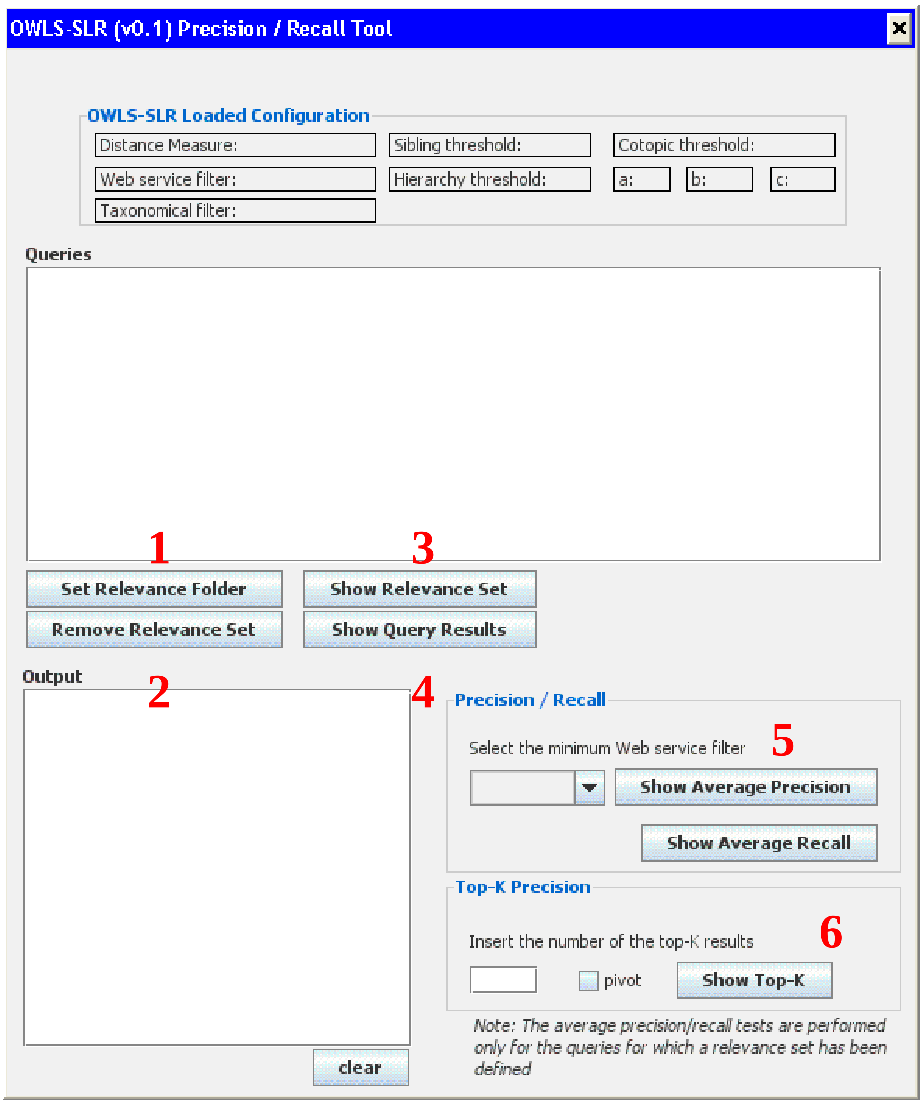
http://127.0.0.1/ontology/RAC.owl

Figure 3. Registering the RAC.owl ontology.

3 Recall and Precision Tool

OWLS-SLR comes with a tool that helps users to determine the recall and precision results of the matchmaker. Of course, the collections should provide the relevance sets of each query in order to be able to perform such kind of experiments. Figure 4 depicts the main window of the tool. More specifically:

1. **Set relevance folder**. Set the relevance folder for a specific query. Note that a query from the list should be first selected.
2. **Remove relevance set**. Removes the registered relevance folder for a specific query.
3. **Show relevance set**. Displays the relevant set of the advertisements for a specific query.
4. **Show query results**. Displays the advertisement results that OWLS-SLR returns for a specific query.
5. **Average Precision/Recall**. Displays the average precision and recall for the queries for which a relevance set has been defined. Users can also define the minimum Web service filter that should be taken into account. For example, if the PLUGIN Web service is selected, then only the advertisements that pass the EXACT and PLUGIN filter will be considered.
6. **Top-K**. Displays the Top-K results, according to the k value. If pivot is selected, then the Top-K results are return for each k value between 1 and k.



Note: The average precision/recall tests are performed only for the queries for which a relevance set has been defined

Figure 4. The precision and recall tool of OWLS-SLR.

4 An Example

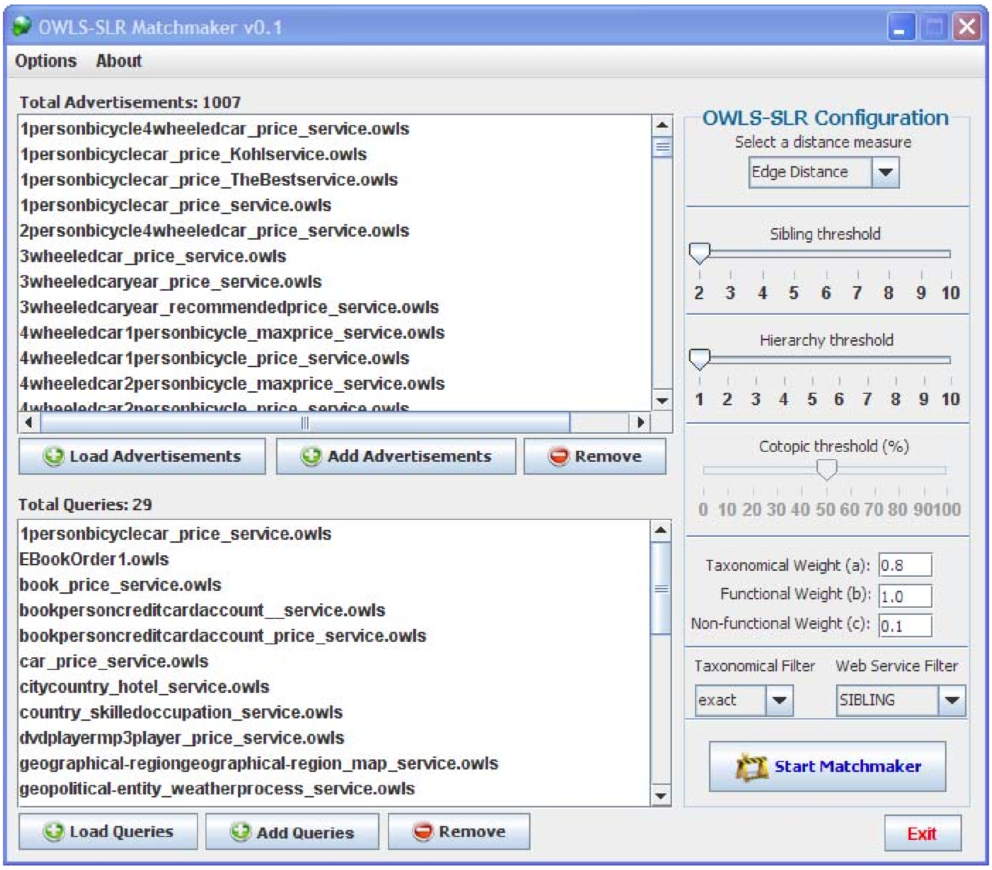
We present an example of loading a dataset and using the precision/recall tool. We will use the OWLS-TC_v2.2_revision_3_(profile_taxonomy) collection [5].

4.1 Registering the taxonomical ontology

Since the specific collection is based on the ProfileTaxonomy.owl ontology, we should firstly register this ontology as a taxonomical, following the procedure we described in section 2.2.

4.2 Loading the Advertisements and Queries

We select the advertisements and the queries from the *services* and *queries* folders (see [5] for dataset installation instructions). Furthermore, we select the SIBLING Web service filter (Figure 5), leaving the other options with their default values.



Total Advertisements: 1007

1personbicycle4wheeledcar_price_service.owls
1personbicyclecar_price_Kohlservice.owls
1personbicyclecar_price_TheBestservice.owls
1personbicyclecar_price_service.owls
2personbicycle4wheeledcar_price_service.owls
3wheeledcar_price_service.owls
3wheeledcaryear_price_service.owls
3wheeledcaryear_recommendedprice_service.owls
4wheeledcar1personbicycle_maxprice_service.owls
4wheeledcar1personbicycle_price_service.owls
4wheeledcar2personbicycle_maxprice_service.owls
4wheeledcar2personbicycle_price_service.owls

Total Queries: 29

1personbicyclecar_price_service.owls
EBookOrder1.owls
book_price_service.owls
bookpersoncreditcardaccount__service.owls
bookpersoncreditcardaccount_price_service.owls
car_price_service.owls
citycountry_hotel_service.owls
country_skilledoccupation_service.owls
dvdplayermp3player_price_service.owls
geographical-regiongeographical-region_map_service.owls
geopolitical-entity_weatherprocess_service.owls

OWLS-SLR Configuration

Select a distance measure: Edge Distance

Sibling threshold: 2 3 4 5 6 7 8 9 10

Hierarchy threshold: 1 2 3 4 5 6 7 8 9 10

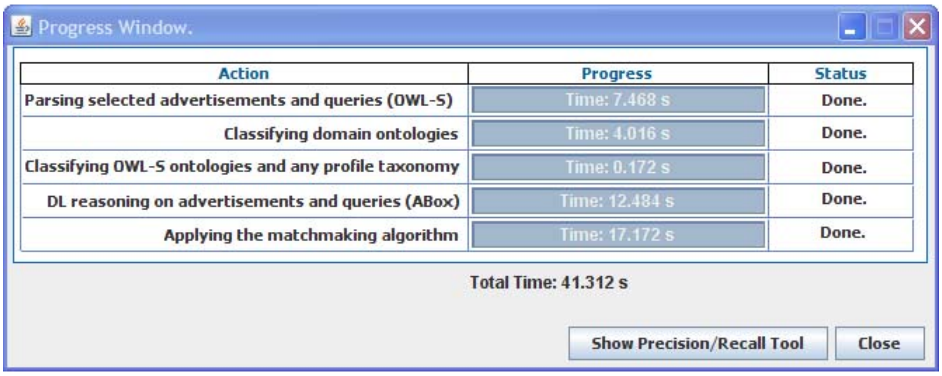
Cotopic threshold (%): 0 10 20 30 40 50 60 70 80 90100

Taxonomical Weight (a): 0.8
Functional Weight (b): 1.0
Non-functional Weight (c): 0.1

Taxonomical Filter: exact
Web Service Filter: SIBLING

Figure 5. Loading advertisements and queries.

By clicking on the Start Matchmaker button, OWLS-SLR starts the processing of the advertisements and queries and the window of Figure 6 is displayed, informing the



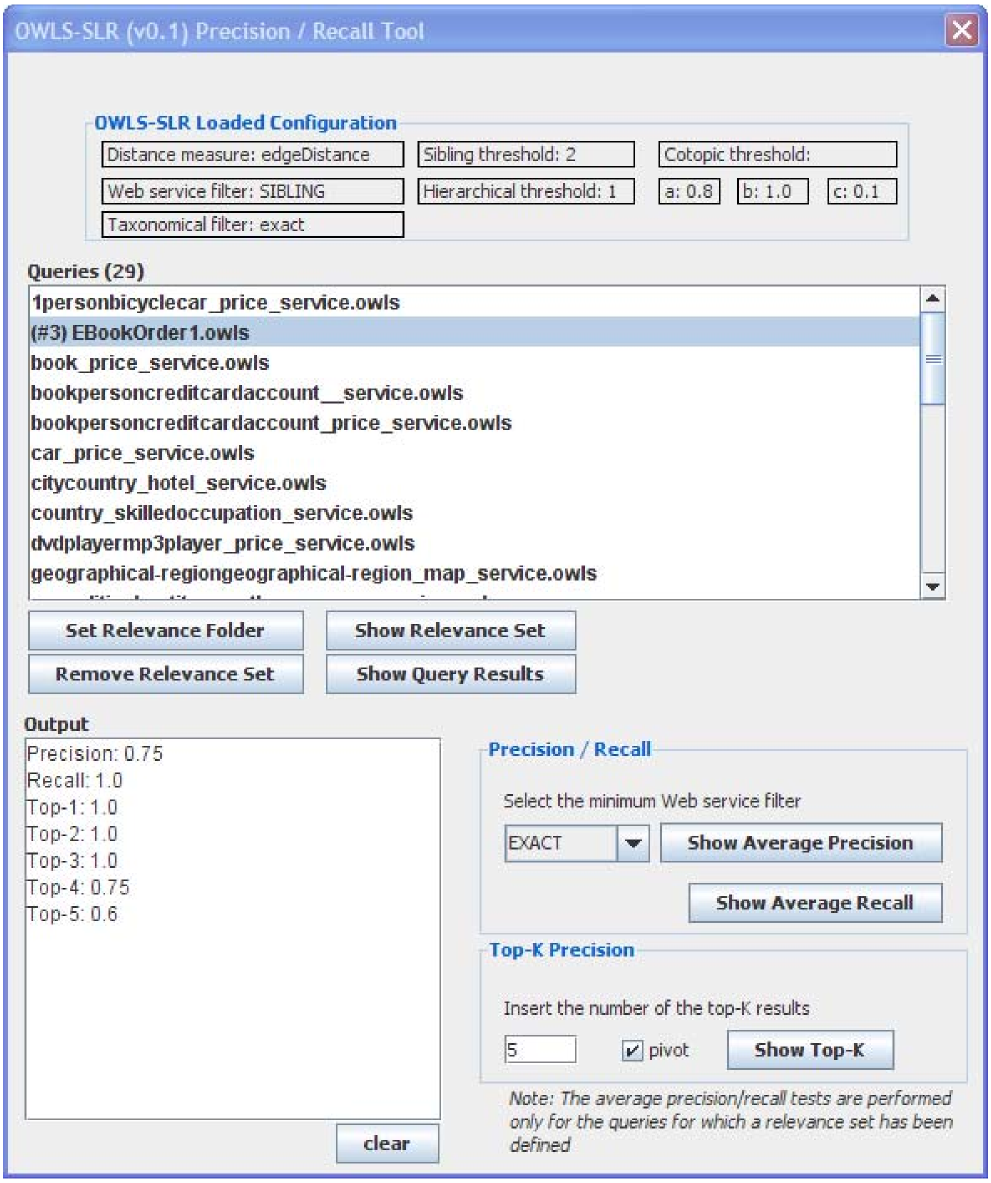
Action	Progress	Status
Parsing selected advertisements and queries (OWL-S)	Time: 7.468 s	Done.
Classifying domain ontologies	Time: 4.016 s	Done.
Classifying OWL-S ontologies and any profile taxonomy	Time: 0.172 s	Done.
DL reasoning on advertisements and queries (ABox)	Time: 12.484 s	Done.
Applying the matchmaking algorithm	Time: 17.172 s	Done.

Total Time: 41.312 s

Figure 6. Progress Window.

user about the progress.

When the matchmaking procedure finishes, the user can click either on the Show Precision/Recall Tool in order to use the precision and recall tool, or on the Close button in order to return to the main window. The precision and recall tool for our example is depicted in Figure 7, where the 29 loaded queries are listed and the user can select the relevance folder of each one by clicking on a specific the query and then on the Set Relevance Folder button. The number of the registered relevant advertisements is depicted in front of the query name, as Figure 7 shows for the EBookOrder1.owls query. The figure shows also the average precision and recall results, as well as the Top-k precision results for k = 5 and the pivot option selected.



Distance measure: edgeDistance

Sibling threshold: 2

Cotopic threshold:

Web service filter: SIBLING

Hierarchical threshold: 1

a: 0.8

b: 1.0

c: 0.1

Taxonomical filter: exact

1personbicyclecar_price_service.owls
(#3) EBookOrder1.owls
book_price_service.owls
bookpersoncreditcardaccount__service.owls
bookpersoncreditcardaccount_price_service.owls
car_price_service.owls
citycountry_hotel_service.owls
country_skilledoccupation_service.owls
dvdplayermp3player_price_service.owls
geographical-regiongeographical-region_map_service.owls

```
Precision: 0.75
Recall: 1.0
Top-1: 1.0
Top-2: 1.0
Top-3: 1.0
Top-4: 0.75
Top-5: 0.6
```

Select the minimum Web service filter

EXACT

Insert the number of the top-K results

5

☑ pivot

Note: The average precision/recall tests are performed only for the queries for which a relevance set has been defined

Figure 7. The precision and recall tool.

References

[1] E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A Practical OWL DL Reasoner,” J. Web Semantics, 2007.

[2] A. Maedche and V. Zacharias, “Clustering Ontology-Based Metadata in the Semantic Web,” European Conf. Principles of Data Mining and Knowledge Discovery, 2002.

[3] R. Bergmann and M. Schaaf, “Structural Case-Based Reasoning and Ontology-Based Knowledge Management: A Perfect Match?,” J. Universal Computer Science, vol. 9, no. 7, 2003.

[4] F. Baader, “The Description Logic Handbook : Theory, Imple-mentation and Applications,” Cambridge University Press, 2003.

[5] OWLS-SLR, http://lpis.csd.auth.gr/systems/OWLS-SLR