Hybrid OWL-S Service Retrieval with OWLS-MX: Benefits and Pitfalls

Matthias Klusch and Benedikt Fries

German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, 66121 Saarbruecken, Germany
klusch@dfki.de

Abstract. The OWLS-MX matchmaker selects OWL-S 1.1 services that are relevant to a given service request by means of combined logic based and approximative matching filters [5]. In this paper, we build upon this work and analyse its retrieval performance in terms of false positive and false negatives to reveal the benefits and pitfalls of both logic based and hybrid matching with OWLS-MX. The analysis results have been exploited in an improved version OWLS-MX+.

1 Introduction

Agent based service discovery aims at coordinating the ultimate service requester with the ultimate service provider agent. This coordination problem can be solved by means of either assisted mediation through middle agents such as matchmakers, brokers and mediators, or in a decentralized peer to peer fashion [4]. In particular, the majority of semantic Web service matchmakers today exploit semantics that are implicit, for example, in patterns or relative frequencies of terms in service descriptions as computed by techniques from data mining, linguistics, or content-based information retrieval.

In line with the recently started shift in semantic Web research towards scalable approximative rather than strict logic based reasoning [2], we proclaim that building semantic service matchmakers purely on DL inferencing, as realized by the majority of existing approaches, might be insufficient in practice. The quality of semantic service discovery can be significantly improved by appropriate exploitation of both crisp logic based and approximate matching where each of them alone would fail.

One example of such a hybrid semantic service matchmaker for OWL-S is OWLS-MX. It takes any desired OWL-S service as a query, and returns an ordered set of relevant services that match the query in terms of both crisp logic based and syntactic similarity. For this purpose, it applies five different hybrid matching filters with one selected token based IR similarity metric each. Logical subsumption failures produced by the integrated OWL-DL reasoner Pellet are tolerated, if the computed syntactic similarity value is sufficient. Experimental evaluation of the performance of hybrid over logic based only matching provided strong evidence in favor of the above claim in terms of both recall and precision

of service retrieval [5]. This evidence is further supported by experimental results reported in [1].

In this paper, we build upon this work and provide an analysis of the retrieval performance of OWLS-MX in terms of false positive and false negatives to reveal the benefits and pitfalls of both its logic based and hybrid matching. The analysis results have been exploited in an improved version of OWLS-MX, called OWLS-MX+, but can be of inherent interest to any developer of hybrid service matchmakers for the semantic Web in general.

The remainder of this paper is structured as follows. Assuming the reader to be familiar with OWL-S, we first summarize the basic idea and hybrid matching filter definitions of OWLS-MX in section 2, followed by a detailed analysis of their false positives and false negatives in section 3. Section 4 briefly presents OWLS-MX+ that has been developed based on the results of this analysis. Finally, section 5 concludes with a summary and open problems.

2 OWLS-MX

The core idea of the OWLS-MX matchmaker is to complement crisp logic based with approximate IR based matching where appropriate to improve the retrieval performance. It takes any OWL-S service as a query, and returns an ordered set of relevant services that semantically match the query each of which annotated with its individual degree of logical matching, and the syntactic similarity value. The user can specify the desired degree, and individual syntactic similarity threshold.

2.1 Matching algorithm overview

OWLS-MX performs signature based service matching only, that is, it compares the input and output parameter values of given OWL-S 1.1 service (query) with those of a given service. More concrete, it first classifies the service query I/O concepts in OWL-DL into its local ontology that contains all I/O concept definitions of advertised services. We assume that the type of computed terminological subsumption relation determines the degree of semantic relation between any pair of I/O concepts.

Second, OWLS-MX then pairwisely determines the degree of logical (concept subsumption) match according to its filter definitions, and the syntactic similarity between the conjunctive service and query I/O concept expressions. These expressions are built by recursively unfolding each query and service input (output) concept in the local matchmaker ontology. The unfolded concept expressions are including primitive components of a basic shared vocabulary only.

Any failure of logical concept subsumption produced by the integrated description logic reasoner of OWLS-MX will be tolerated, if and only if the degree of syntactic similarity between the respective unfolded service and query concept expressions exceeds a given similarity threshold. The detailed matching algorithm with a brief example are given in [5].

2.2 Matching filters

Let T be the terminology of the OWLS-MX matchmaker ontology specified in OWL-DL; CT_T the concept subsumption hierarchy of T; $LSC(C)$ the set of least specific concepts (direct children) C' of C, i.e. C' is immediate sub-concept of C in CT_T; $LGC(C)$ the set of least generic concepts (direct parents) C' of C, i.e., C' is immediate super-concept of C in CT_T; $Sim_{IR}(A, B) \in [0, 1]$ the numeric degree of syntactic similarity between strings A and B according to chosen IR metric IR with used term weighting scheme and document collection, and $\alpha \in [0, 1]$ given syntactic similarity threshold; $\doteq$ and $\dot{\geq}$ denote terminological concept equivalence and subsumption, respectively. The matching filters of OWLS-MX are as follows.

Exact match. Service S EXACTLY matches request R $\Leftrightarrow \forall\ \text{IN}_S\ \exists\ \text{IN}_R$: $\text{IN}_S \doteq \text{IN}_R \wedge \forall\ \text{OUT}_R\ \exists\ \text{OUT}_S$: $\text{OUT}_R \doteq \text{OUT}_S$.

Plug-in match. Service S PLUGS INTO request R $\Leftrightarrow \forall\ \text{IN}_S\ \exists\ \text{IN}_R$: $\text{IN}_S \dot{\geq} \text{IN}_R \wedge \forall\ \text{OUT}_R\ \exists\ \text{OUT}_S$: $\text{OUT}_S \in \text{LSC}(\text{OUT}_R)$. Relaxing the exact matching constraint, service S may require less input than it has been specified in the request R. This guarantees at a minimum that S will be executable with the provided input iff the involved OWL input concepts can be equivalently mapped to WSDL input messages and corresponding service signature data types. We assume this a necessary constraint of each of the subsequent filters. In addition, S is expected to return more specific output data whose logically defined semantics is exactly the same or very close to what has been requested by the user.

Subsumes match. Request R SUBSUMES service S $\Leftrightarrow \forall\ \text{IN}_S\ \exists\ \text{IN}_R$: $\text{IN}_S \dot{\geq} \text{IN}_R \wedge \forall\ \text{OUT}_R\ \exists\ \text{OUT}_S$: $\text{OUT}_R \dot{\geq} \text{OUT}_S$. Compared to the plug-in filter the constraint of immediate output concept subsumption is relaxed. As a consequence, the returned set of relevant services is extended in principle.

Subsumed-by match. Request R is SUBSUMED BY service S $\Leftrightarrow \forall\ \text{IN}_S\ \exists\ \text{IN}_R$: $\text{IN}_S \dot{\geq} \text{IN}_R \wedge \forall\ \text{OUT}_R\ \exists\ \text{OUT}_S$: $(\text{OUT}_S \doteq \text{OUT}_R \vee \text{OUT}_S \in \text{LGC}(\text{OUT}_R)) \wedge \text{SIM}_{IR}(\text{S, R}) \geq \alpha$. This filter selects services whose output data is more general than requested, hence, in this sense, subsumes the request. We focus on direct parent output concepts to avoid selecting services returning data which we think may be too general.

Logic-based fail. Service S fails to match with request R according to the above logic-based semantic filter criteria.

Nearest-neighbor match. Service S is NEAREST NEIGHBOR of request R $\Leftrightarrow \forall\ \text{IN}_S\ \exists\ \text{IN}_R$: $\text{IN}_S \dot{\geq} \text{IN}_R \wedge \forall\ \text{OUT}_R\ \exists\ \text{OUT}_S$: $\text{OUT}_R \dot{\geq} \text{OUT}_S \vee \text{SIM}_{IR}(\text{S, R}) \geq \alpha$.

Fail. Service S does not match with request R according to any of the above filters.

The above filters are in the following total order according to the size of results they would return, i.e. according to how relaxed the semantic matching:

$$\text{EXACT} < \text{PLUG-IN} < \text{SUBSUMES} < \text{SUBSUMED-BY} < \text{LOGIC-BASED FAIL} < \text{NEAREST-NEIGHBOR} < \text{FAIL}.$$

2.3 OWLS-MX variants

We implemented the following variants of OWLS-MX, called OWLS-M1 to OWLS-M4, each of which uses the same logic-based semantic filters but different (token based) IR similarity metric $\text{SIM}_{IR}(R, S)$ for content-based service I/O matching. The variant OWLS-MO performs logic based only semantic service I/O matching.

OWLS-M0. The logic-based semantic filters EXACT, PLUG-IN, and SUBSUMES are applied as defined in 2.2, whereas the hybrid filter SUBSUMED-BY is utilized without checking the syntactic similarity constraint.

OWLS-M1 to OWLS-M4. The hybrid semantic matchmaker variants OWLS-M1, OWLS-M2, OWLS-M3, and OWLS-M4 compute the syntactic similarity value SIM_{IR} (OUT_S, OUT_R) by use of the loss-of-information measure (M1), extended Jaccuard similarity coefficient (M2), the cosine similarity value (M3), and the Jensen-Shannon information divergence based similarity value (M4), respectively.

2.4 Implementation

The OWLS-MX matchmaker has been implemented in Java using the OWL-S API 1.1 beta with the tableaux OWL-DL reasoner Pellet developed at the university of Maryland (cf. `http://pellet.owldl.com/`). As the OWL-S API is tightly coupled with the Jena Semanic Web Framework, developed by the HP Labs Semantic Web research group (cf. `http://jena.sourceforge.net/`), the latter is also used to modify the OWLS-MX matchmaker ontology.

The OWLS-MX matchmaker in its current version 1.1c is accessible via a convenient graphical user interface. It also provides a module (OWLS-MXP) for service I/O compatibility checking on the WSDL grounding level based on respective XMLS data type checking. OWLS-MX is available as open source from the portal semwebcentral.org [1].

3 R/P Performance Analysis

In this section, we first provide an overview of the R/P performance of the OWLS-MX variants followed by an analysis of false positives and false negatives returned by OWLS-M0 and the hybrid variants. That is, we are interested in typical cases where logic based (OWLS-M0) matching benefits from complementary approximative reasoning (OWLS-M1 to OWLS-M4) on the description of service semantics, where it fails, and vice versa.

[1] http://projects.semwebcentral.org/projects/owls-mx/

3.1 Overall R/P performance of OWLS-MX

For measuring the service retrieval performance of each OWLS-MX variant we used the OWL-S service retrieval test collection OWLS-TC v2 [2], and adopted the evaluation strategy of micro-averaging the individual recall/precision (R/P) curves. The micro-averaged R/P curves of the top and worse performing IR similarity metric together with those for the OWLS-MX variants are shown in figure 3.1.

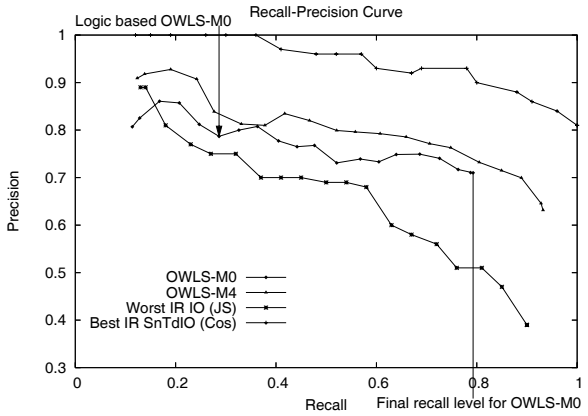

Fig. 1. R/P performance of OWLS-MX variants

This preliminary result reported in [5] provides strong evidence in favor of the proposition that building semantic Web service matchmakers purely on crisp logic based reasoning may be insufficient.

Quantitative analysis A preliminary quantitative analysis of these results in [5] showed that even the best IR similarity metric (Cosine/TFIDF) alone performed close to the pure logic based OWLS-M0 which can be significantly outperformed by hybrid semantic matching with OWLS-M1 to OWLS-M4 in terms of both recall and precision. Second, the hybrid matchmakers OWLS-MX, in turn, can be outperformed by each of the selected syntactic IR similarity metrics to the extent additional parameters with natural language text content are considered.

At this point, we add that the cosine IR similarity metric and OWLS-M4 using the Jensen-Shannon similarity metric performed best for syntactic and hybrid service profile I/O matching, respectively. Though, all hybrid OWLS-MX variants showed almost equal performance in average. These experimental results, of course, largely depend on the service retrieval test collection used.

[2] Available at `http://projects.semwebcentral.org/projects/owls-tc/`.

Building upon this preliminary quantitative analysis of the overall R/P performance, we continued our evaluation experiments to determine the main reasons of the false positives and false negatives of the logic based OWLS-M0 in more detail. In the following sections, we summarize the results of the respective qualitative analysis.

3.2 Logic based false positives

The qualitative analysis of irrelevant services returned by OWLS-M0 (false positives lower precision) in the experiments revealed the following characteristic reasons of their occurence in its answer sets.

1. The subsumption based distance between concepts in the matchmaker ontology insufficiently captures their real-world semantics (RWS) to be detected by logic based OWL-M0 filters. In contrast, a hybrid matching filter can mitigate this problem by determining the syntactic similarity between the respective concept expressions.
2. The surjective mapping of I/O concepts by M0 filters tolerates missing of concepts that are key for description of service semantics.
3. Same I/O concepts are used with inherently different semantics which cannot be detected by OWLS-M0.

These types of logic based false positives of OWLS-M0 are illustrated by example in the following.

Granularity of matchmaker ontology Of course, if the concept subsumption (or parent-child) relations in the ontology insufficiently capture the corresponding real-world semantics (RWS) any logic based filter risks to produce false positives.

For example, in figure 3.2, the service at best plug-in matches with the query, since their equally named output concepts "price" are equivalent, and the query input "HybridRotaryEnginePoweredCar" is far more specific than that of the considered service, that is "Automobile". In fact, the semantic distance between both input concepts in the ontology is probably too large for being of any interest to the user in practice. The reason why all logic based matching filters of OWLS-M0 fail in such a case is that they accept an unlimited input concept distance.

On the other hand, the subsumed-by filter of the hybrid variants like OWLS-M2 showed a better performance in these cases as the low syntactic similarity between the unfolded concept expressions treated as strings indicates a high degree of irrelevance. The relevant parts of the logical filter definitions are marked in red: first one is Plug-in(S,Q) "service S plugs into query Q", second is subsumes(Q,S) "query Q subsumes service S", third is subsumed-by(Q,S) "query Q is subsumed by the service S" with LSC and LGC denoting the set of least specific (direct child) and generic (direct parent) concepts.

The second example shown in figure 3.2 refers to the case of service and query output concepts with close distance in a coarse-grained matchmaker ontology but

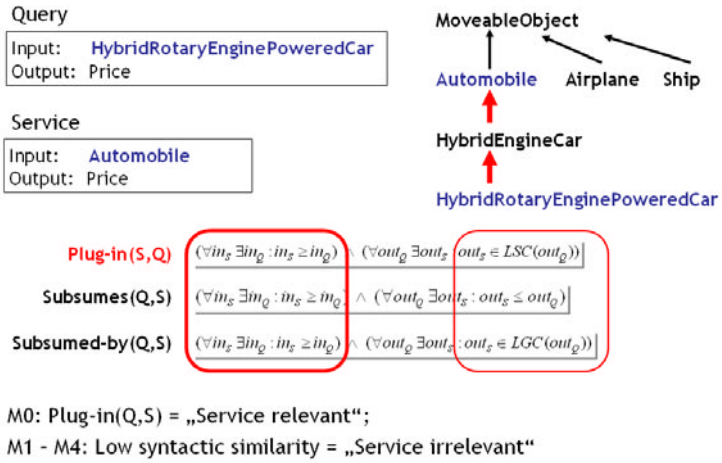

Fig. 2. Example: Unlimited input concept distance accepted by OWLS-M0

high RWS distance. In this example, even a least generic concept match of the logic based Subsumed-By(Q,S) filter of OWLS-M0 does not help.

Surjective mapping of concepts Another reason of false positives produced by the logic based OWLS-M0 is due to the surjective mapping of service and query concepts. In fact, the service and query input concepts might not be correlated by logical filters though they are carrying inherent semantics.

In figure 3.2, for example, the input "SFNovel" of the query "SFNovelPrice" does not match with any input of the service "EntranceFee" but "Author" with "Person". As a result, OWLS-M0 determines a plug-in(S,Q) or Subsumed-By(Q,S) match, hence the service relevant to the query though it definitely is not. The reason of this type of false positive is that the surjective mapping of service input concepts by OWLS-M0 filters tolerates the missing of concepts that are key for the description of query semantics.

Incomplete coverage of service semantics Another type of logic based false positives is caused by the insufficient coverage of service semantics by the definitions of I/O concepts used.

For example, the real world semantics of service "BookCopyCheck" and query "BookReview" in figure 3.2 are not related at all but both are determined by OWLS-M0 as semantically equivalent. The concept "Book" is used twice in the service input but with different semantics than in the query. In these cases, even our syntactic similarity measurements returned high relevance but at least not identity.

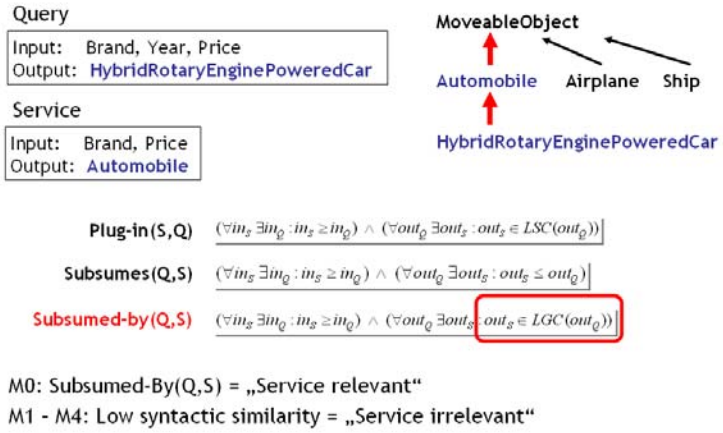

Fig. 3. Example: Close output concepts with high RWS distance

3.3 Mitigating logic based FP problems

All FP problems described above call for the complementary use of text retrieval similarity metrics with fine-grained syntactic overlap measurement. In fact, our experimental results show that the number of false positives can be drastically reduced by using hybrid filters - which leads to a significant improvement in precision as shown in figure 3.3 for the case of applying OWLS-M0 vs OWLS-M2 to the respective test collection.

However, the same experiments also revealed that even hybrid OWLS-MX variants return false positives which we briefly illustrate next.

3.4 False positives of hybrid matching

Main reasons for hybrid OWLS-MX variants to return false positives are that all filters (a) do not require a total mapping of I/O concepts between service and query, and (b) used syntactic similarity measurements ignore the semantics of logical operators in concept expressions. In fact, our experiments showed that adding services of the above types of FP to the OWLS-TC2 causes a significant decrease in the precision of all OWLS-MX variants.

Surjective mapping of concepts According to their definition, no filter of OWLS-MX requires a total (bijective) mapping of service and query I/O concepts which can lead to false positives in situations where no concepts are provided .

For example, in figure 3.4, the query "BuyBook" and service "DatingService" are returned as equivalent since there is no query output concept to be matched.

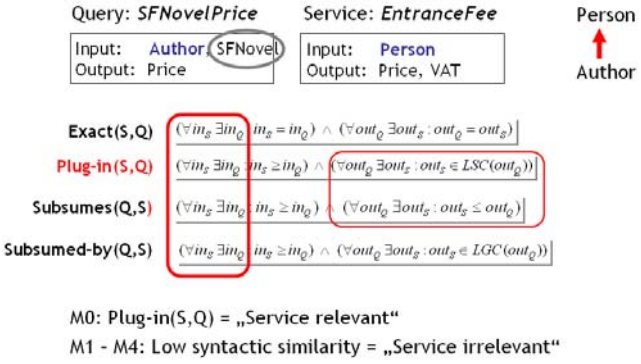

Fig. 4. Example: Tolerated missing of key concepts for service or query semantics

Logical operator semantics Another reason of false positives returned by hybrid variants is that the used syntactic similarity means ignore the logical language operators in concept definitions. In fact, connectives like "and", "or" are eliminated as stop words in the preprocessing step of IR similarity measurement. For example, in figure 3.4, the query "B/W-Print" is asking for print outs in black or white, while the service "BW-Print" is offering print outs in black and white. The service is logically irrelevant but identical for syntactic similarity metrics.

3.5 Logic based false negatives

The experimental analysis of relevant services missed by the logic based matchmaker OWLS-M0 (false negatives lower recall) revealed the following main characteristic reasons of failure.

1. In fine-grained matchmaker ontology, service concepts can logically differ from those of a given query but have similar real-world semantics (e.g. semantically close siblings in the ontology).
2. Opposed to the all-quantified matching filter constraints of OWLS-M0, the pairwise comparison of service and query I/O concepts does not always yield the same type of subsumption relationship.
3. In OWLS-M0, all service input concepts have to be equal or more generic than provided as it would, in case of a linear mapping of concepts to WSDL grounding data types, guarantee service invocation but may be too restrictive in certain cases.

Concept siblings Logic based false negatives can be caused by logically disjoint definitions of sibling concepts with similar real-world semantics. For example, in figure 3.5, the real world semantics of query output "Hopital-Physician" and the service output "Emergency-Physician" are quite similar.

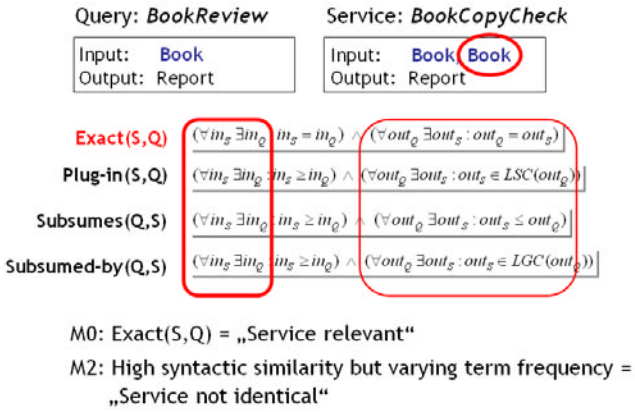

Fig. 5. Example: False positive due to incomplete coverage of semantics

In the extreme, only one pair of conjunctive logical constraints or primitive components of siblings in the concept hierarchy remains unmatched during subsumption computation, hence causes a logical failure by OWLS-M0. Since both unfolded concept expressions are syntactically identical otherwise, any hybrid OWLS-MX would return both concepts as similar with respect to their implicit semantics exploited by selected IR similarity metric

Different subsumption relations Another major class of logic based false negatives is caused by the all-quantified matching filter constraints of OWLS-M0 that requires each pair of service and query I/O concepts to have the same type of subsumption relation.

For example, in figure 3.5, the types of subsumption relations between the output concepts of query "CarPlusBike" and service "4WheeledCarPackage" are different. As a result, the logic based OWLS-M0 fails while the hybrid variants determin the service as relevant, if the degree of the syntactic similarity between the considered pairs of concepts is sufficient.

Enforcement of more generic service input Finally, in OWLS-M0, all service input concepts have to be equal or more generic than provided which might not be the case though the service is relevant to the given query as shown in figure 3.5

3.6 Mitigating logic based FN problems

As mentioned above, both kinds of identified FN problems call for the complementary use of text retrieval similarity metrics with fine-grained syntactic overlap measurement.

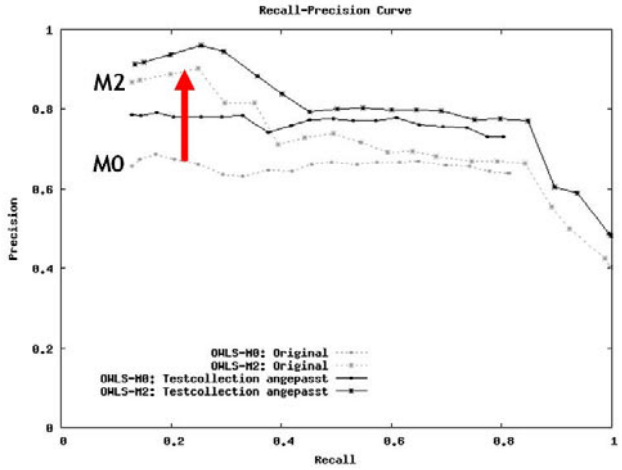

Fig. 6. Mitigating logic based false positives with hybrid matching filters that exploit syntactic similarity measurement.

The experimental results depicted in figure 3.6 show that the recall of all hybrid variants is significantly better than the one of OWLS-M0 applied to a test collection with sets of services that cause logic based false negatives. Main reason for this is, that the additional IR based similarity check of the nearest-neighbor filter allows OWLS-M1 to -M4 to find relevant services that the logic based OWLS-M0 would fail to retrieve.

4 Advancing OWLS-MX retrieval performance

Regarding the results of the FP/FN analysis of the retrieval performance of OWLS-MX in the previous section, we are interested in how most of the identified reasons of failure could be overcome. Basic idea is to improve the R/P performance by allowing for an integrated rather than just complementary hybrid matching on the deeper level of service I/O concepts.

That is, the respectively updated matchmaker version OWLS-MX+ determines whether

- a total (bijective) mapping of service/query I/O concepts exists, and
- further relaxes the match by tolerating also individual pairs of service/query concepts (instead of considering their input and output as a whole) with their logical relation deviating from that required by the all-quantification filtering constraints, if their syntactic similarity is sufficient.

The experimental results show that the R/P performance of OWLS-MX+ is significantly improved over that of OWLS-MX (cf. figure 4). Please note that OWLS-MX+ is not available in the latest version 1.1c of OWLS-MX; we are currently working on the next release (OWLS-MX 2.0) where it will be integrated.

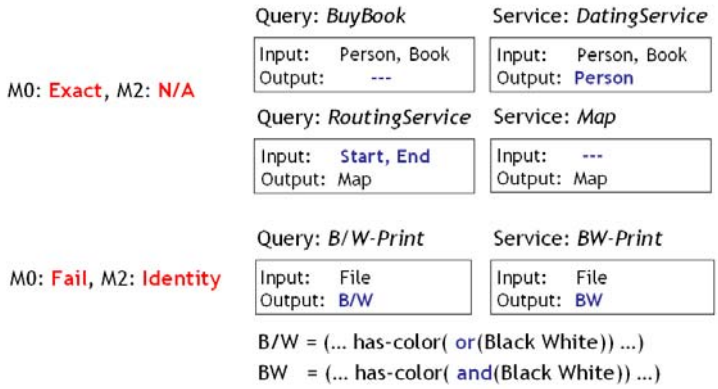

Fig. 7. Example: False positives of hybrid matching

5 Conclusions

Since its first release in May 2005, the OWLS-MX matchmaker is the only hybrid matchmaker for OWL-S services that applies both logic based and approximative matching filters. For an accessible account of other approaches to semantic Web service discovery, we refer to, for example, [3].

The results of our experimental analysis of hybrid semantic matching filters of OWLS-MX showed that the main characteristic problems of logic based only false positive/negatives can be largely mitigated by (a) syntactic overlap measurements of IR metrics, and (b) integrated hybrid matching on deeper concept level improves precision (OWLS-MX+).

However, one open problem of OWLS-MX+, as for any other logic based SWS matchmaker, is that the importance of individual concepts used to define the real world semantics of the considered service is highly subjective thus may vary for different users. This is hard to deal with in general. The problem of syntactic similarity metrics ignoring the semantics of logical operators "and", "or" in the expanded concept definitions as classical stop words will be solved in an updated version of OWLS-MX 2.0.

Finally, the query relevance sets of the used collection OWLS-TC2 are, of course, highly subjective. The construction of a large scale service retrieval collection has to be a joint effort of many people within and outside the SWS community. We encourage the reader to contribute to the evolution and enlargement of the OWLS-TC online at the sws-tc wiki http://www-ags.dfki.uni-sb.de/swstc-wiki

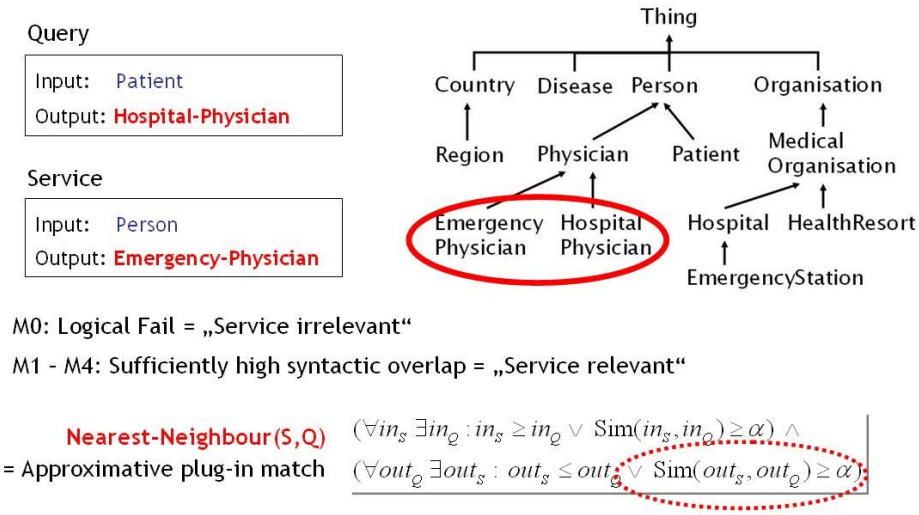

Fig. 8. Example: False negative for concept siblings in the ontology

References

1. A. Bernstein, C. Kiefer: Imprecise RDQL: Towards Generic Retrieval in Ontologies Using Similarity Joins. Proc. ACM Symposium on Applied Computing, Dijon, France, ACM Press, 2006.
2. D. Fensel, F. van Harmelen: Unifying reasoning and search to Web scale. *IEEE Internet Computing*, March/April 2007.
3. S. Grimm. Discovery - Identifying Relevant Services. In: R. Studer, S. Grimm, A. Abecker (eds.). Semantic Web Services. Springer, 2007.
4. M. Klusch, K. Sycara: Brokering and Matchmaking for Coordination of Agent Societies: A Survey. In: Coordination of Internet Agents: Models, Technologies and Applications. Springer, 2001.
5. M. Klusch, B. Fries, and K. Sycara. Automated Semantic Web Service Discovery with OWLS-MX. Proceedings of 5th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Hakodate, Japan, 2006
6. OWLS-MX: http://projects.semwebcentral.org/projects/owls-mx/.
7. OWLS-TC: http://projects.semwebcentral.org/projects/owls-tc/.

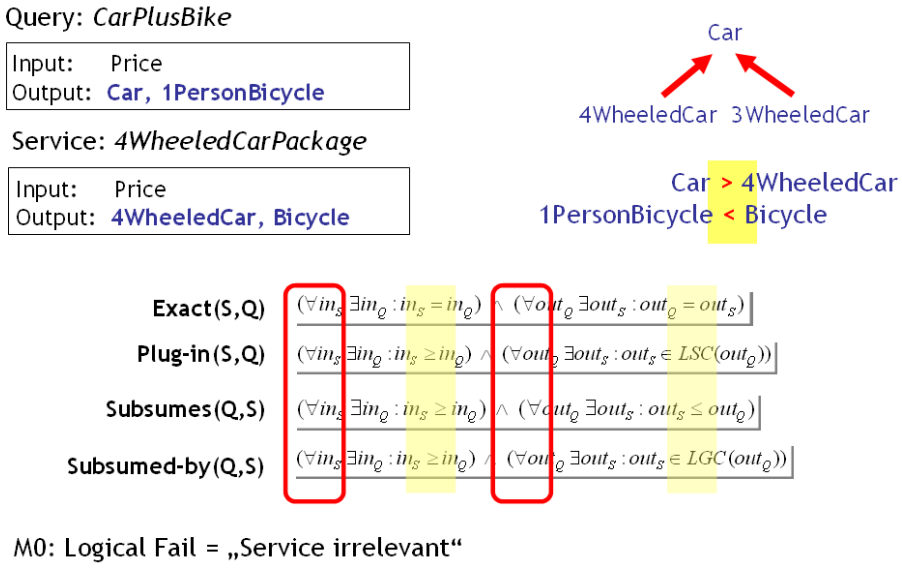

Fig. 9. Example: False negative due to different concept subsumption relations

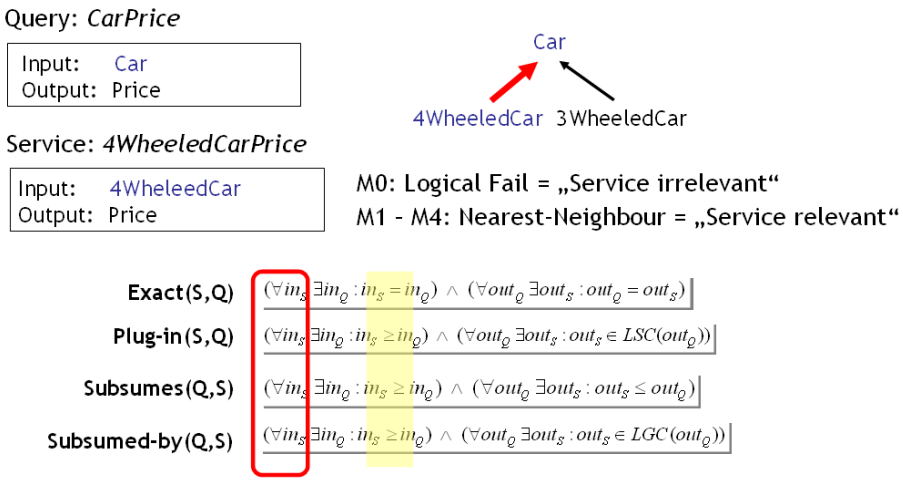

Fig. 10. Example: False negative due to required more generic service input

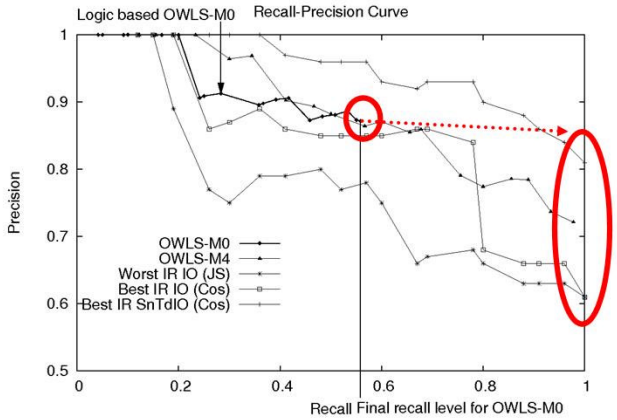

Fig. 11. Mitigating false negative problems of logic based OWLS-M0 with hybrid matching filters

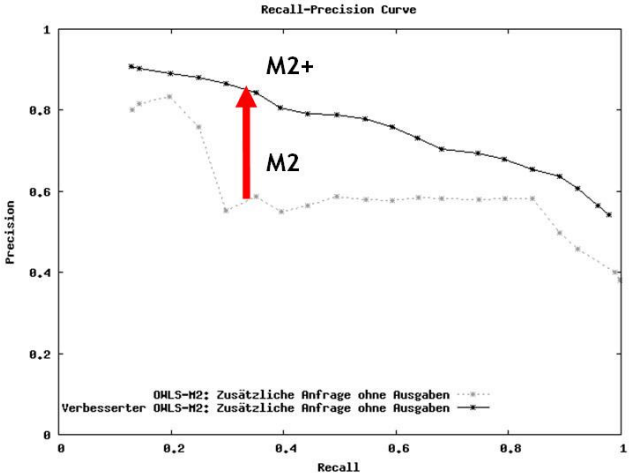

Fig. 12. Improved R/P performance with OWLS-MX+