OWLS-MX3: An Adaptive Hybrid Semantic Service Matchmaker for OWL-S

Matthias Klusch and Patrick Kapahnke

German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, Saarbrücken, Germany
klusch|patrick.kapahnke@dfki.de

Abstract. We present OWLS-MX3, the first adaptive hybrid semantic Web service matchmaker for OWL-S. It learns how to best combine logic-based, text similarity and ontological structure matching for hybrid semantic selection of OWL-S services to given queries. For this purpose, the matchmaker utilizes a SVM-based classifier which is learned over a training set of the test collection OWLS-TC3 off-line. In particular, it performs structural semantic matching to compensate for certain cases of text matching failures caused by the observed characteristic of many Semantic Web ontologies today of being rather mere is-a ontologies. Our comparative retrieval performance evaluation experiments based on standard measures for both binary and graded service relevance revealed a rather negative result: There is a slight but not significant improvement of performance over the non-adaptive variant OWLS-MX2 yet. On the other hand, its adaptation feature clearly renders OWLS-MX3, in principle, independent from any OWL-S test collection with referenced ontologies and any kind of matching filters that are to be properly (re-)combined by hand otherwise to reflect changes in the environment.

1 Introduction

At the core of each discovery of relevant services in the semantic Web is the process of semantic selection that can be automatically performed by so-called semantic service matchmaker. Semantic service selection comprises the process of matching and ranking of a given pair of service request and offer. In this paper, we present the first adaptive hybrid semantic OWL-S service matchmaker, called OWLS-MX3. Its development is in line with our work on hybrid semantic service selection techniques for prominent service description formats such as SAWSDL [11], OWL-S [12] and WSML [10]. In particular, it is based on the experimental evaluation results for the hybrid service matchmaker OWLS-MX2 [13] which utilizes a fixed combination of logic-based and text similarity-based matching techniques and may significantly outperform each of its single matching techniques in practice.

Hybrid semantic matching as performed by OWLS-MX2 has its own deficiencies, in particular text matching failures which can be avoided by additional structural concept matching as we will show in this paper. Besides, the problem of how to

best combine different kinds of semantic service matching in a way that renders the matchmaker independent from both its actual service collection and any set of matching filters to be used in combination with reasonable performance in terms of average precision and recall.
For this purpose, we propose the first adaptive hybrid semantic OWL-S service matchmaker named OWLS-MX3 that learns to optimally aggregate the results of different matching filters by utilizing a binary SVM-based classifier trained over a given test collection such as OWLS-TC[1]. In this respect, OWLS-MX3 follows the idea of our adaptive matchmaker SAWSDL-MX2[11] for SAWSDL services - but significantly differs from it in the kind of structural matching performed, and, of course, in the service description format.
Both, fixed (non-adaptive) and adaptive hybrid semantic OWL-S service matchmakers OWLS-MX2, respectively, OWLS-MX3 shall then be compared to each other with respect to their quantitative retrieval performance in practice. Finally, we also compare these experimental evaluation results with those obtained for both of them over a new, more fine-grained test collection OWLS-TC3 with both graded and binary relevance sets.
The remainder of this paper is structured as follows. In section 2, for readers not familiar with OWLS-MX2, we briefly recall its complimentary integration of logic-based and text similarity-based matching. Section 3 then shows how text matching failures of OWLS-MX2 can be avoided by structural matching with respect to specific characteristics of the majority of ontologies used for semantic service annotation in the Semantic Web today. Based on these results, we then present the first adaptive hybrid matchmaker for OWL-S services, that is OWLS-MX3, in section 4. Results of comparative experimental performance evaluation are provided in section 5, while a brief discussion of related works is in section 6. We conclude the paper in section 7.

2 Recall: Fixed Hybrid Semantic Matchmaker OWLS-MX2

Key to the hybrid semantic OWL-S service matchmaker OWLS-MX2 is that it avoids logic-based service signature matching failures by means of a fixed hybrid combination of logical with approximate matching based on text similarity measurement. In order to understand and compare both hybrid OWL-S matchmaker variants, the fixed OWLS-MX2 and the adaptive OWLS-MX3, we briefly present the hybrid selection techniques of the first one, that is its means of semantic service matching and ranking. For more details on OWLS-MX2 together with a case study and examples, we refer the interested reader to [13, 14]. OWLS-MX2[2] combines means of logical with text similarity-based matching of OWL-S service signatures.

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

Logic-based matching. The crisp logic-based matching filters of OWLS-MX2 define its logical variant OWLS-M0 and are subsequently applied to a given pair of service request and service offer in OWL-S until one of these filters evaluates to true. Let be LSC(C) the set of least specific concepts (direct children) of C, and LGC(C) the set of least generic concepts (direct parents) of C in the concept subsumption graph; and in:C $\in Input_S$ (out:C $\in Output_S$) an input (output) concept C of service S defined in the shared ontology. Then the logical concept subsumption-based filters of OWLS-MX2, in other words its pure logical variant OWLS-M0, are defined as follows:

Exact match. Service S EXACTLY matches request R $\Leftrightarrow \forall$ in:C $\in Input_S$ $\exists$ in:C'$\in Input_R$: C $\equiv$ C' $\wedge$ $\forall$ out:D$\in Output_R$ $\exists$ out:D'$\in Outputs_S$: D $\equiv$ D'.

Plug-in match. Service S PLUGS INTO request R $\Leftrightarrow \forall$ in:C$\in Input_S$ $\exists$ in:C'$\in Input_R$: C' $\sqsubseteq$ C $\wedge$ $\forall$ out:D$\in Output_R$ $\exists$ out:D'$\in Outputs_S$: D'$\in$ LSC(D);

Subsumes match. Request R SUBSUMES service S $\Leftrightarrow \forall$ in:C$\in Input_S$ $\exists$ in:C'$\in Input_R$: C' $\sqsubseteq$ C $\wedge$ $\forall$ out:D$\in Output_R$ $\exists$ out:D'$\in Outputs_S$: D' $\sqsubseteq$ D.

Subsumed-by match. Request R is SUBSUMED BY service S $\Leftrightarrow \forall$ in:C$\in Input_S$ $\exists$ in:C'$\in Input_R$: C' $\sqsubseteq$ C $\wedge$ $\forall$ out:D$\in Output_R$ $\exists$ out:D'$\in Outputs_S$: D' $\equiv$ D $\vee$ D'$\in$ LGC(D).

Logical Fail. OWLS-MX returns a logic-based semantic matching failure degree, iff service S does not match with request R according to any of the above matching filters.

These matching degrees are sorted according to the order of semantic relevance degrees as follows: Exact < Plug-In < Subsumes < Subsumed-By < Logical Fail [13]. Our initial experimental evaluation of OWLS-M0 over the publicly available test collection OWLS-TC2 showed that many logical matching failures could be avoided by additional standard text similarity-based matching.

Text similarity-based matching. Based on the work of Cohen and his colleagues [3], we selected the following top performing, symmetric token-based text similarity measures (X) for OWLS-MX2: The intensional loss-of-information ($X = LOI$) metric, and the vector-space model TFIDF-based Cosine and Tanimoto (Extended Jaccard, EJ) coefficients. The input and output concepts of service offer and request are represented as text depending on the text similarity measure and model used. OWLS-MX2 also computes the text similarity $sim_{ann,X}$ of the informal annotation of OWL-S services. TFIDF weighting is applied using distinct term index for inputs, output, respectively informal annotation. The overall syntactic similarity $\textsc{Sim}_{IR}$(S, R) between service S and request R is computed as the average of the two syntactic similarity values $sim_{InConc,X}$, $sim_{OutConc,X}$ of their inputs, respectively, outputs, and $sim_{ann,X}$. Different variants of OWLS-MX2 use the same logic-based semantic filters but different text similarity metrics[3].

[3] Variant OWLS-M12 computes the LOI-based syntactic similarity, while OWLS-M22 and OWLS-M32 compute the Cosine, respectively, the Tanimoto (extended Jaccard) coefficient.

Non-adaptive hybrid combination of logical and text matching. Our evaluation experiments with OWLS-MX2 over OWLS-TC2 showed that both logical and text matching are sufficiently statistically independent from each other. This justifies the use of both matching conditions in conjunction (except for the degree EXACT) with strict condition of sufficiently high text similarity of semantic signatures. This can avoid logical false positives, while the final non-logic-based nearest-neighbor matching degree (Service S is NEAREST NEIGHBOR of request R ⇔ $\textsc{Sim}_{IR}$(S, R) $\geq \alpha$) can avoid logical false negatives. The hybrid service matching degrees are sorted as before with NEAREST-NEIGHBOR < FAIL in addition.
Our experiments with OWLS-MX2 over test collection OWLS-TC2 [14, 13] revealed that its hybrid semantic matchmaking may outperform both logic-based and text similarity-based service selection. On the other hand, the latter kind of matching may introduce misclassifications by its own which can be avoided by structural semantic matching.

3 Structural Semantic Matching

The structural matching of semantic signatures of services S and R relies on the concept similarity measure proposed by Li et al. [16]:

$$sim_{dist}(C_R, C_S) = \begin{cases} e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & , CR \neq C_S \\ 1 & , C_R = C_S \end{cases} ,$$

where l denotes the shortest path distance between the concepts C_R and C_S in the given is-a ontology T, h is the depth of their *direct common subsumer* in T, α and β are parameters weighting the importance of l and h respectively. In line with the analysis in [16], $\alpha = 0.2$ and $\beta = 0.6$ provided good results in the evaluation. This concept similarity measure was especially designed by Li et al. (2003) to represent the intuitive meaning of similarity depending on the level of abstraction of concepts to be compared in a given taxonomy, i.e. the distances of more specialized concept definitions reflect the similarity more significant than distances of upper concept definitions (e.g. animal/plant vs. dog/wolf)[16].
To compute a single structural matching value based on this structural concept similarity measure, we apply the same principle to the sets A, respectively, B of input (or output) concepts of services S, respectively, R:

$$sim_C(A, B) = \frac{1}{|A|} \cdot \sum_{a \in A} max\{sim_{dist}(a, b) | b \in B\},$$

That is, the overall structural similarity of two concept sets is the average of the maximum concept similarities found for concepts in B for each concept in A. This definition allows the computation of two distinct matching values for the input and output parts of two service signatures. In addition, we take the number of signature parameters for service offer and request into account. This is motivated by the fact that our experimental evaluation over the test collection OWLS-TC revealed that the all-quantified logical matching constraint in OWLS-MX may cause logical false positives in case the number of parameters of service offer and request differ. To avoid this, the parameter checking function $sim_M(A,B) = 1 - \frac{||A|-|B||}{max\{|A|,|B|\}}$, is applied to the input and output part of the service signatures separately, and used in the following structural input, respectively, output similarity functions for service offer S and request R:

$$sim_{S,in}(R,S) = \gamma \cdot sim_C(S_{in}, R_{in}) + (1-\gamma) \cdot sim_M(S_{in}, R_{in}),$$

$$sim_{S,out}(R,S) = \gamma \cdot sim_C(R_{out}, S_{out}) + (1-\gamma) \cdot sim_M(R_{out}, S_{out}),$$

with weight $0 \leq \gamma \leq 1$ (for OWLS-TC2, $\gamma = 0.5$ turned out to be best). Finally, the overall structural matching value for both service signatures is defined as the average of their structural input and output similarities:

$$sim_{struct}(R,S) = \frac{sim_{S,in}(R,S) + sim_{S,out}(R,S)}{2}.$$

In mere is-a ontologies, this structural concept similarity-based matching can indeed help to avoid logical and text misclassifications as illustrated by the simple example in figure 1. In this case, the structural similarity of the service offer and request output concepts ”GraduateSchool” and ”Organization” is low, as it only focuses on their structural relation (path distance) via direct common subsumer in the ontology - in contrast to both text and logical matching each of which bases on the complete (and quite overlapping) concept unfoldings, hence return a false positive semantic matching result.
Similarly, the logical and text false negatives in the example of figure 2 is caused by comparing two concepts (”Airport” and ”RailwayTerminal”) of not related services which unfoldings in the ontology vary significantly due to the multiple inheritance of a large subtree by one of them (”Airport”). However, their distance via direct common subsumer concept with depth 6 is low (=2), and their structural similarity matching value ($sim_{struct}(R,S)$) is sufficiently high for S being classified as relevant.

4 Adaptive Hybrid Semantic Service Selection

As mentioned abvoe, the design of the hybrid semantic matching filters of OWLS-MX2 based on the results of an exhaustive evaluation analysis [13]. The problem with this non-adaptive approach is how to avoid repeating this tedious analysis every time the set of services used for matching a given request changes? How to best combine different matching filters to obtain a reasonable retrieval performance in terms of precision and recall? Inspired by the work of [7] and [9], we developed the adaptive hybrid semantic matchmaker OWLS-MX3 that simply learns how to resolve this problem by learning a support vector machine-based classifier for its service selection off-line. This renders it independent from any service collection and kind of matching filter to be integrated in the future.

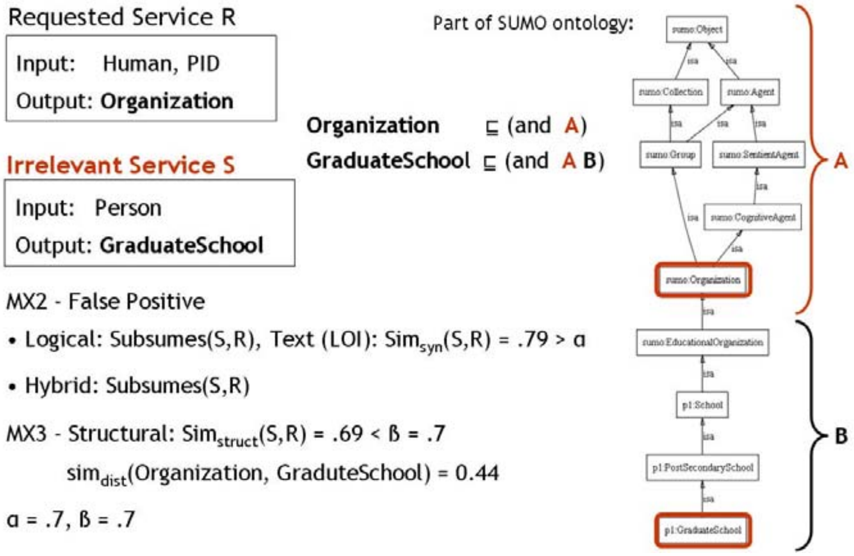

Fig. 1. Avoiding logical and text false positives by structural matching.

4.1 OWLS-MX3 Overview

In short, the OWLS-MX3 matchmaker returns a ranked list of relevant services S for a given request R in OWL-S based on the aggregated results of separately performed logical, text and structural similarity-based matching. Each of these different matching filters has been described above. Their aggregation by OWLS-MX3 is optimal with respect to average classification accuracy according to its binary SVM-classifier that has been learned over a training set in prior. In the following, we describe the off-line learning and use of the SVM-classifier for hybrid semantic service selection by OWLS-MX3 in more detail.

4.2 SVM Classifier for Service Selection

Learning of binary SVM classifier. The problem of classifying a given service S with respect to its semantic relevance to a given request R can be re-formulated as the problem of learning a binary support vector machine-based classifier. That is, how to find a separating hyperplane for a given feature space X such that for all positive and negative training samples with minimal distances (support vectors) to it, these distances are maximal. In case of OWLS-MX3, we consider a 7-dimensional feature space $X = \{0,1\}^5 \times [0,1] \times [0,1]$, where each of the first five binary dimensions corresponds to the occurrence of one out of five different logical matching degrees (*exact, plug-in, subsumes, subsumed-by, fail*) followed by the two real-valued dimensions for text, respectively, structural similarity-based matching degrees. For example, the feature vector $x_i = (0, 0, 1, 0, 0, 0.6, 0.7)$

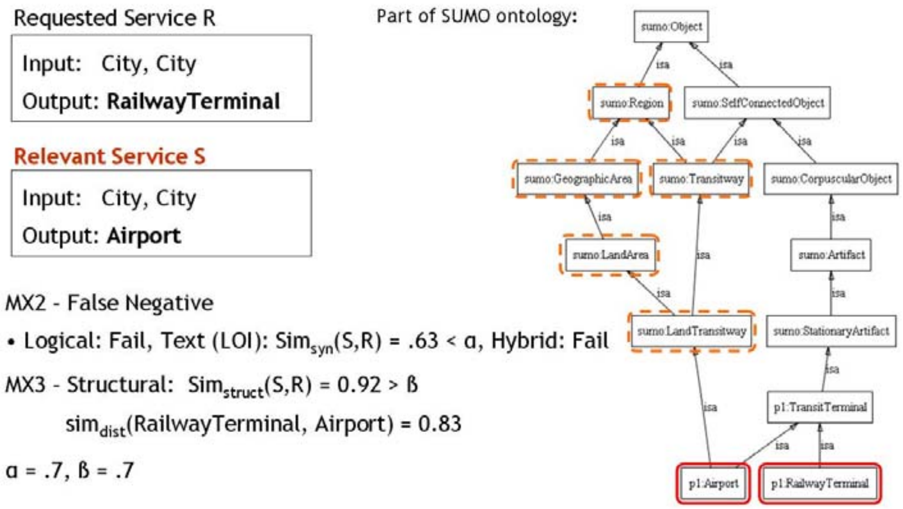

Fig. 2. Avoiding logical and text false negatives by structural matching.

($i \leq N$, N is the size of the training set of positive and negative samples) indicates that the matching results for the service offer/request pair (S,R) that corresponds to the training sample (x_i, y_i) (with $y_i = 1$ iff S is relevant to R according to the binary relevance sets defined in OWLS-TC3, else $y_i = -1$) yields a logical *subsumes* match, a text similarity of 0.6, and structural similarity of 0.7. For the training set $\{(x_1, y_1) \ldots (x_N, y_N)\}$, we randomly selected 700 samples with equal quantities of positive and negative relevance samples. This amounts to around 20% of the complete search space of samples (which size is the number of requests times the number of services in OWLS-TC3) over which the binary SVM classifier for service relevance is learned. The SVM classification problem is defined as the following optimization problem: minimize in w,b,ζ: $\frac{1}{2} w^T w + C \sum_{i=1}^{N} \zeta_i$, subject to $\forall 1 \leq i \leq N : y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \zeta_i \geq 0$, where w and b define the optimally separating hyperplane as the set of points satisfying $w^T \phi(x) + b = 0$. Furthermore, w is the *normal vector* which specifies the orientation of the plane, b is called *bias* and indicates the offset of the hyperplane from the origin of the feature space X. The error term $C \sum_{i=1}^{N} \zeta_i$ is introduced to allow for outliers in a non-linear separable training set, where the error penalty parameter C must be specified beforehand. The predefined function ϕ maps features into a higher, possibly infinitely dimensional space in which the SVM finds a hyperplane that allows a classification of non-linear separable data (more precise with respect to the original dimension of X)[4].

Since $w = \sum_{i=1}^{N} y_i \alpha_i \phi(x_i)$ is a linear combination of training sample feature vectors the dual formulation of the SVM classification problem that is actually

[4] The fraction $\frac{1}{2}$ is introduced for computational reasons only, and does not affect the classification result.

solved by OWLS-MX3 is as follows: maximize in α: $\frac{1}{2}\sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i$, subject to $\sum_{i=1}^{N} y_i \alpha_i = 0, \forall 1 \leq i \leq N : 0 \leq \alpha_i \leq C$. The *kernel* function $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ implicitly defines ϕ in the scalar product, while problem is solved by finding a set of Lagrange multipliers α_i representing the hyperplane for which training samples x_i with $\alpha_i \neq 0$ are called support vectors (of the hyperplane). For OWLS-MX3, we choose the RBF-Kernel (Radial Basis Function) $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$ as suggested in [8]. Unlike polynomial kernels, it only introduces a single parameter γ which keeps the complexity of model selection low. Besides, for specific parameter settings it can behave like a linear or sigmoid kernel.
The searching of an optimal SVM parameter setting (C, γ) with respect to average classification accuracy has been done through means of grid search and 6-folded cross-validation. Binary classification of samples $x \in X$ for service pair (S,R) with the above mentioned parameters is defined as follows: $d(x) = \sum_{i=1}^{N} y_i \alpha_i K(x_i, x) + b$ with bias b satisfying the Karush-Kuhn-Tucker condition (KKT)[1], such that S is classified as relevant iff $d(x) > 0$. Please note, that w is not a direct output of the dual optimization but computed using the objective value o of the dual optimization and the coefficients α_i based on the relation between the primary and dual problem: $||w||^2 = w^T w = \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j) = 2 \cdot (o + \sum_{i=1}^{N} \alpha_i)$.

Use of trained SVM classifier for selection. The semantic selection of relevant services S to any request not in the training set then bases applying the learned binary classifier d to the corresponding matching feature vector x of (S,R) as described above: S is relevant to R, if and only if $d(x) > 0$, otherwise it is classified as irrelevant. The service S is then ranked according to the distance $dist(x) = \frac{d(x)}{|w|} \in R^+$ of its feature vector x to the learned hyperplane of the classifier such that the hybrid semantic matching degree eventually returned by OWLS-MX3 for (S,R) is the tuple $(d(x), dist(x))$.

5 Experimental Evaluation of Performance

One key question regarding the adaptive hybrid matchmaker OWLS-MX3 presented above is whether it can improve the retrieval performance of its fixed hybrid counterpart OWLS-MX2? We did perform a comparative evaluation of the performance in terms of standard information retrieval performance measurement for both matchmakers together with the logical OWLS-M0, and the Tanimoto text similarity-based service selection over the test collection OWLS-TC3 with both binary and graded service relevance assessments.

5.1 Experimental Setting

The matchmaker OWLS-MX3 has been fully implemented in Java using the reasoner Pellet for OWL-DL reasoning, and libSVM[5] for its SVM module im-

[5] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

plementation.

Test collection OWLS-TC3. The test collection OWLS-TC3 consists of 1007 service offers and 29 service requests in OWL-S covering different application domains such as travel, tourism and e-health. It has been derived from the test collection OWLS-TC2[6] mainly through (a) WSDL services as groundings for all OWL-S services in the collection, and (b) additional graded relevance sets for its queries based on appropriate collaborative user-based service relevance rating. There is no other OWL-S test collection publicly available yet. OWLS-TC is an ongoing joint effort of different institutions and widely used by the community. All ontologies used for its real-world semantic services are publicly available such as the SUMO and the Mid-level ontology. The fact that most ontologies in the Semantic Web today (which are also used in OWLS-TC3) are rather is-a ontologies still hampers the use of any logic-based matchmaker in principle. That is not a weakness of the collection but, by contrast, reflects its compliance with reality in this respect. However, please note that OWLS-TC3 is still far from being a standard collection comparable to TREC in the information retrieval domain, but that certainly shall not put research and development of semantic service matchmakers with preliminary performance evaluation on hold in general until such a collection might become available in the future.

The matchmaker performance tests have been conducted on a machine with Windows 2000, Java 6, 1,7 GHz CPU and 2 GB RAM using the latest version 2.0 of our semantic service matchmaker evaluation environment SME^2 which is available at projects.semwebcentral.org/projects/sme2/.

Retrieval performance measures based on binary relevance. Binary retrieval performance is usually measured in terms of well-known measures of *recall* and *precision*: $Rec = \frac{|A \cap B|}{|A|}, Prec = \frac{|A \cap B|}{|B|}$, where A is the set of all relevant documents for a request and B the set of all retrieved documents for a request. For our evaluation, we adopt the prominent macro-averaging of precision, that is the mean of precision values for answer sets returned by the matchmaker for all queries in the test collection at standard recall levels Rec_i $(0 \leq i < \lambda)$. Ceiling interpolation is used to estimate precision values that are not observed in the answer sets for some queries at these levels. The number of recall levels from 0 to 1 (in equidistant steps $\frac{n}{\lambda}, n = 1 \ldots \lambda$) we used for our experiments is $\lambda = 20$. The macro-averaged precision is defined as follows: $Prec_i = \frac{1}{|Q|} \cdot \sum_{q \in Q} \max\{P_o | R_o \geq Rec_i \wedge (R_o, P_o) \in O_q\}$, where O_q denotes the set of observed pairs of recall/precision values for query q when scanning the ranked services in the answer set for q stepwise for true positives. Besides, we measure the single value of *Average Precision* (AP) for each matchmaker variant for a single query q: $AP_q = \frac{1}{|R|} \sum_{r=1}^{|L|} isrel(r) \frac{count(r)}{r}$, where R is the set of relevant items previously defined by domain experts for the given query q, L the ranking of returned items for that query, $isrel(r) = 1$ if the item at rank r

[6] available at projects.semwebcentral.org/projects/owls-tc/

grade	gain value	explanation
highly relevant	3	The service offer perfectly satisfies the service request.
relevant	2	Any service offer that answers the request completely or at least partially, but has additional conditions that are not fulfilled completely.
partially relevant	1	Any service offer that may be helpful to fulfill the request (e.g. by providing related information).
non-relevant	0	The service offer is not relevant to the request at all.

Table 1. Grading scale and assigned gain values.

is relevant and 0 otherwise and $count(r)$ the number of relevant items found in the ranking when scanning top-down, i.e. $count(r) = \sum_{i=1}^{r} isrel(i)$. The overall average precision AP of a matchmaker then averages the AP values over all queries in the test collection. The AP measure enables performance evaluation that is invulnerable to varying sizes of ranked lists of services returned by different matchmakers. [17] propose a variant AP' of the AP measure for condensed lists to cope with incomplete relevance sets. In this case, which is in particular valid for the test collection OWLS-TC3 (due to lack of human resources for relevance judgments), not every service offer/request pair has been judged or rated (by different users) with respect to the semantic relevance of the offer to the request. A condensed ranking is a ranking, where all entries are dropped which are not part of the set of relevance assessments. Since the standard macro-averaged precision measure cannot be applied to incomplete relevance sets as is, every service offer in an unrated service offer/request pair is treated as *irrelevant*. For our experiments, inspired by TREC[18], all service offer/request pairs from the top-100 results of various matchmakers that participated in the S3 service selection contest in 2009 are rated.

Retrieval performance measures based on graded relevance. Since the OWLS-TC3 also provides a more fine-grained service relevance assessment by introducing different grading scales, we also measured the performance of the matchmakers according to the respective measures for graded instead of binary relevance as investigated in [17]. For the grading scale, we selected the one given in table 1 which is taken from the NTCIR project (see [2]). According to [17] the performance measures based on these graded relevance assessments are robust to the choice of gain values, thus we settled for the linear decrement given in the second column (the intuitive meaning of each of the grades is indicated in the right part of the table).

The performance based on graded relevance is measured in terms of both the Q and the $nDCG$ measure both of which rely on accumulated gain values. The Q-measure is defined as follows: $Q = \frac{1}{|R|} \sum_{r=1}^{|L|} isrel(r) \frac{\beta cg(r) + count(r)}{\beta cg_I(r) + r}$, where $cg(r)$ denotes the *cumulative gain* at rank r, i.e. $cg(r) = \sum_{1 \leq i \leq r} g(i)$ with $g(i)$ being the gain value for the retrieved document at rank i, $cg_I(r)$ is the cumulative gain value of an ideal ranked output and β controls the penalty on lower ranked

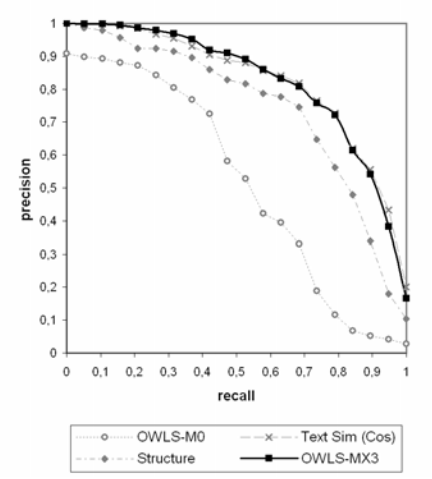

Fig. 3. Macro-averaged Recall/Precision for hybrid OWLS-MX3 vs logical OWLS-M0 vs text vs structural matching.

relevant documents. Since the robustness of the Q-measure to the choice of β has been shown, we set $\beta = 1$ (please also note that $\beta = 0$ reduces the Q-measure to AP). To apply this measure to incomplete relevance sets, condensed ranking lists are considered for evaluation forming Q'. The $nDCG$ measure is based on *discounted* gain values $dg(r) = \frac{g(r)}{\log_a(r)}$ for $r > a$ and $dg(r) = g(r)$ for $r \leq a$. Analogous to the definition of cg_I, $dg_I(r)$ denotes the discounted gain for a hypothetic perfectly ranked list. The overall measure then is defined as follows: $nDCG_l = \sum_{r=1}^{\max(|L|,l)} dg(r) / \sum_{r=1}^{l} dg_I(r)$, where l is a predefined cut-off value. For our experiments, $l = 100$ has been chosen, because too small cut-offs may hurt the stability of nDCG (see [17]). We set the logarithmic base $a = 2$, and apply variant $nDCG'$ using condensed lists to incomplete relevance sets in OWLS-TC3 as recommended in [17].

5.2 Performance Based on Binary Relevance

The results of our experimental performance evaluation of OWL-S matchmakers based on binary relevance sets in OWLS-TC3 are summarized in figure 3 and 4.

The learned hybrid aggregation of OWLS-MX3 performs significantly better in terms of precision than most of the basic matchmaker variants except for the text similarity approach using the TFIDF *Cosine* measure - which even outperforms the logical OWLS-M0 [12]. This is due to the fact that most Semantic Web ontologies, hence those in the collection used for service annotations, appear to be rather simple in terms of mere is-a taxonomies. For example, the standard SUMO

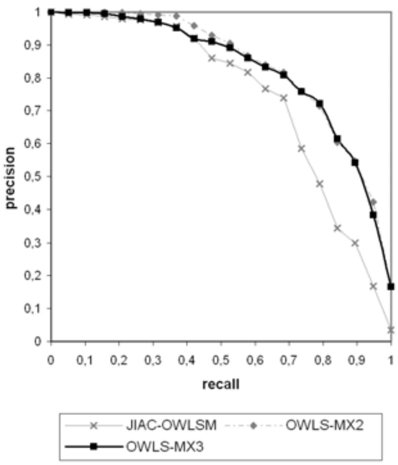

Fig. 4. Macro-averaged R/P for adaptive hybrid OWLS-MX3 vs fixed hybrid OWLS-MX2 and JIAC-OWLSM.

and Mid-level[7] ontologies mainly define concepts in terms of concept inclusion axioms, hence do not exploit the full expressivity of description logics such as OWL-DL and the implicit knowledge respectively inferred by a DL reasoner. One weakness of pure text matching to neglegt the logical operators in concept definitions does not significantly decrease its performance in practice yet.

In figure 4, the fixed hybrid semantic service selection by OWLS-MX2 is compared to the machine-learning approach of OWLS-MX3. Unfortunately, none of both performs significantly better than the other, that is, their retrieval performance does not significantly differ according to the statistical Friedman Test performed on the AP' values for each query resulting in $p \approx 0.182$ (means, there is no significant advantage of one approach over the other at 5% level [5]). An additional AP' test resulted in approximate average values of 0.86 for OWLS-MX2 and 0.84 for OWLS-MX3, which confirms the results of the macro-averaged recall/precision analysis.

We also added the JIAC-OWLSM[8] to get some comparative evaluation results, which is also plotted in figure 4. JIAC-OWLSM performs hybrid semantic service signature matching based on subsumption relations between service offer and request I/O concepts mapped to numerical scores, as well as simple text similarity (string equality/containment). A fixed weighting scheme is applied to compute the overall similarity value. JIAC-OWLSM was one of the participants of the *Semantic Service Selection* (S3) contest[9] in 2008. As can be seen, both OWLS-MX2 and OWLS-MX3 outperform JIAC-OWLSM, especially at late re-

[7] http://www.ontologyportal.org/

[8] TU Berlin, DAI Lab

[9] http://www-ags.dfki.uni-sb.de/∼klusch/s3/

	OWLS-M0	text sim.	structure	OWLS-MX2	OWLS-MX3	JIAC-OWLSM
avg. Q'	0.74	0.83	0.79	**0.85**	0.84	0.71
avg. nDCG'	0.82	0.9	0.87	0.9	**0.91**	0.85
avg. QRT	2.9 sec	3.2 sec	2.0 sec	5.6 sec	7.1 sec	22.1 sec

Table 2. Evaluation results based on graded relevance, and query response time.

call levels (significantly at 5% level according to the Friedman test ($p \approx 0.0012$) and with average AP' value of 0.74.

5.3 Performance Based on Graded Relevance

The fixed hybrid matchmaker OWLS-MX2 did perform best in terms of the Q'-measure incorporating the more fine-grained, that is graded semantic relevance assessments. However, it turned out, that the discrepancy of most matchmaker variants is smaller than for the binary relevance-based performance measures except for the crisp logic-based approach of OWLS-M0. This is in perfect line with the investigation in [15] which, in particular, states that an evaluation performance analysis with graded relevance appears more appropriate for semantic service matchmakers.

Interestingly, the top performer in the evaluation changes to OWLS-MX3 for the nDCG'-measure which discounts the worth of correct late retrievals. That is, OWLS-MX3 seems to perform sligthy more precise for the top ranking positions, though the discrepancy is insignificant at 5% level according to the statistical Friedman test ($p \approx 0.362$). Besides, in some case the evaluation results for graded relevance assessments turn to the opposite obtained for those with binary relevance assessments, e.g., the performance results for binary relevance as shown in figure 4 impose a different supposition. Finally, the OWLS-MX outperformed the alternative hybrid OWL-S matchmaker JIAC-OWLSM in both cases of relevance assessment.

Another standard performance measure in text IR is the average query response time which results are shown in table 2. The mere structure-based selection is the fastest variant followed by the other basic matching approaches. The fixed hybrid OWLS-MX2 takes approximately as long as the logical OWLS-M0 and text similarity-based selection together - which is not surprising since the filter design requires both kinds of computations for every service pair, though the logical classification of service request I/O concepts has only to be performed once. The adaptive OWLS-MX3 is most slow in its query response due to (a) its additional structural matching filter, and (b) the additional off-line training phase. The generated feature vector for each request/offer pair has to be tested against the trained SVM, which is in $O(n)$ for n being the number of support vectors forming the seperating hyperplane. JIAC-OWLSM is significantly slower than any OWLS-MX variant, but its service offer registration phase is quite fast since concept classification for offers is done at query time only (in contrast to OWLS-MX).

6 Related Work

To the best of our knowledge, there does not exist any other adaptive hybrid semantic matchmaker for OWL-S services yet. The closest work to OWLS-MX3 is the one, called SAWSDL-MX2, for adaptive selection of SAWSDL services which has recently been presented in [11]. SAWSDL-MX2 differs from OWLS-MX3 in both the service description format and the completely different kind of structural matching based on XMLS tree similarity of WSDL documents ignoring its semantic annotations. The OWLS-iMatcher [9] integrates various text similarity measures applied to OWL-S service descriptions by means of different machine-learning algorithms. As with the OWLS-MX3, its performance evaluation over OWLS-TC2 showed that its adaptive aggregation can significantly outperform each of the selected text similarity measurements in terms of precision. This is in perfect line with the evaluation results presented by Cohen et al. [3] on SVM-based combination of different text similarity metrics. In the context of adaptive Web search, Joachims et al. [7] presented an integrated meta-search engine, called Striver, that improves its precision by learning over basic features extracted from previous search results using a SVM classifier. This work particularly inspired our work on OWLS-MX3 for adaptive semantic Web service retrieval. Regarding evaluation strategies of semantic Web service matchmaking in general, Küster et al. [15] extensively discuss the advantages of more fine grained test collections with graded instead of binary relevance assessments. This further motivated our updating of the test collection OWLS-TC2 with binary relevance sets to OWLS-TC3 including also graded relevance sets, and the updating of our semantic service matchmaker evaluation environment SME2 2.0 that offers standard evaluation measurements for both cases of relevance assessment.

7 Conclusions

We presented an adaptive hybrid semantic OWL-S service matchmaker, called OWLS-MX3, that learns how to best combine different matching filters, that are logical, text and structural matching, in terms of precision and recall. For this purpose, it aggregates the separate matching results according to a trained binary SVM classifier and ranks the relevant services accordingly. The results of our experimental performance evaluation over the test collection OWLS-TC3 showed that OWLS-MX3 is competitive to the fixed hybrid OWL-S service matchmaker OWLS-MX2 regarding precision but still not significantly better. However, the main benefit of using OWLS-MX3 is that its adaptive aggregation of different kinds of matching results renders it, in principle, independent from any test collection with its referenced ontologies in the changing Semantic Web in the future - while, by contrast, any fixed matchmaker such as OWLS-MX2 would have to be adapted manually by the developer to reflect such changes, if required. OWLS-MX3 is planned to be released in November 2009.

References

1. Chang, CC.; Lin, CJ.: LIBSVM: a library for support vector machines. Available at http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001
2. Chen, K. et al.: Overview of CLIR task at the third NTCIR workshop.
3. Cohen, WW.; Ravikumar,P.; Fienberg, SE.: A comparison of string distance metrics for name-matching tasks. Proceedings of IIWeb conference, 2003
4. Hsu, CW.; Chang, CC.; Lin, CJ.: A Practical Guide to Support Vector Classification. 2007
5. Hull, D.: Using statistical testing in the evaluation of retrieval experiments. Proceedings of 16th annual international ACM SIGIR conference on Research and development in information retrieval, 1993
6. Joachims, T.: Optimizing Search Engines Using Clickthrough Data. Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.
7. Joachims, T.; Radlinski, F.: Search Engines that Learn from Implicit Feedback. IEEE Computer, 40(8), 2007
8. Keerthi, SS.; Lin, CJ.: Asymptotic behaviour of support vector machines with Gaussian kernel. Journal on Neural Computation, 15(7), 2003
9. C. Kiefer, A. Bernstein: The Creation and Evaluation of iSPARQL Strategies for Matchmaking. Proceedings of 5th European Semantic Web Conference (ESWC), Lecture Notes in Computer Science, Vol. 5021, 463–477, Springer-Verlag Berlin Heidelberg, 2008
10. Klusch, M.; Kaufer, F. (2009): WSMO-MX: A Hybrid Semantic Web Service Matchmaker. Journal of Web Intelligence and Agent Systems, 7(2), IOS Press
11. Klusch, M.; Kapahnke, P.; Zinnikus, I. (2009): Hybrid Adaptive Web Service Selection with SAWSDL-MX and WSDL Analyzer. Proceedings of 6th European Semantic Web Conference (ESWC), Heraklion, Greece, IOS Press.
12. Klusch, M.; Fries, B.; Khalid, M., Sycara, K. (2005): OWLS-MX: Hybrid Semantic Web Service Retrieval. Proceedings of the 1st International AAAI Fall Symposium on Agents and the Semantic Web, Arlington VA, USA, AAAI Press, 2005
13. Klusch, M.; Fries, B.; Sycara, K. (2009): OWLS-MX: A Hybrid Semantic Web Service Matchmaker for OWL-S Services. Journal of Web Semantics, 7(2), Elsevier
14. Klusch, M., Fries, B.: Hybrid Semantic Web Service Retrieval: A Case Study with OWLS-MX. Proceedings of 2nd IEEE International Conference on Semantic Computing (ICSC), Santa Clara, USA, IEEE Press, 2008
15. Küster, U.; König-Ries, B.: Evaluating Semantic Web Service Matchmaking Effectiveness Based on Graded Relevance. Proceedings of the 2nd International Workshop SMR^2 on Service Matchmaking and Resource Retrieval in the Semantic Web at the 7th International Semantic Web Conference (ISWC08), 2008
16. Y. Li, A. Bandar, D. McLean: An approach for measuring semantic similarity between words using multiple information sources. Transactions on Knowledge and Data Engineering 15, 2003
17. T. Sakai: Alternatives to Bpref. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007
18. E. M. Voorhees: Overview of trec 2002. Proceedings of the 11th Text Retrieval Conference (TREC), NIST Special Publication 500-251, 2002