Chapter 1
Self-Adaptive Semantic Matchmaking using COV4SWS.KOM and LOG4SWS.KOM

Ulrich Lampe and Stefan Schulte

Abstract This chapter presents the methodological and technical approach, as well as evaluation results, for two semantic matchmakers, COV4SWS.KOM and LOG4SWS.KOM. Both matchmakers operate on WSDL-based service description with SAWSDL annotations. COV4SWS.KOM applies similarity measures from the field of semantic relatedness, namely the metrics by Lin and Resnik. It automatically adapts to varying expressiveness of a service description on different abstraction levels through the utilization of an Ordinary Least Squares (OLS) estimator. LOG4SWS.KOM employs traditional subsumption reasoning, but maps the resulting discrete Degrees of Match (DoMs) to numerical equivalents to allow for the integration with additional similarity measures. As proof of concept, a path length-based measure is applied. The DoM mapping process may either be conducted manually or using an OLS estimator. Both matchmakers participated in the Semantic Service Selection (S3) Contest in 2010, providing very competitive evaluation results across all regarded performance metrics.

1.1 Introduction

In the envisioned *Internet of Services*, (Web) services will be commodities that are traded via public marketplaces. One important prerequisite to realizing this vision consists in effective and efficient service discovery, i.e., the ability to find (functionally) matching services based on a given query. In current research, this discovery process is commonly not only based on syntactical, but also on semantic information, as provided by *Semantic Web Services* (SWS) [17].

Ulrich Lampe
Technische Universität Darmstadt, Germany, e-mail: ulrich.lampe@kom.tu-darmstadt.de

Stefan Schulte
Technische Universität Darmstadt, Germany, e-mail: stefan.schulte@kom.tu-darmstadt.de

In the chapter at hand, we introduce COV4SWS.KOM and LOG4SWS.KOM[1], two semantic matchmakers for WSDL-based service descriptions with SAWSDL annotations[2]. Both matchmakers participated in the *Semantic Service Selection* (S3) Contest in 2010, achieving very favorable results in terms of the regarded Information Retrieval (IR) metrics [10].

COV4SWS.KOM and LOG4SWS.KOM are based on a common framework, named XAM4SWS ("Cross-Architectural Matchmaker for Semantic Web Services"). However, in the treatment of semantic annotations and the implementation of self-adaptiveness, we pursue different approaches in both matchmakers.

COV4SWS.KOM applies similarity measures from the field of semantic relatedness, namely the metrics by Lin [14] and Resnik [25]. It automatically adapts to varying expressiveness of a service description on different abstraction levels through the utilization of an Ordinary Least Squares (OLS) estimator [30].

LOG4SWS.KOM employs traditional subsumption reasoning, but maps the resulting discrete *Degrees of Match* (DoMs) to numerical equivalents. This allows for the direct integration with additional (numerical) similarity measures. As proof of concept, a path length-based measure is applied. The DoM mapping process may either be conducted manually or using an OLS estimator.

Common features of both matchmakers include the use of a rudimentary fallback strategy, based on the WordNet English language ontology [18]. Also, the principal methodology of determining service similarities is identical. Namely, this concerns the use of bipartite graph matching and the aggregation of similarity values from different service abstraction levels.

1.2 Approach: COV4SWS.KOM and LOG4SWS.KOM

1.2.1 Common Characteristics

While COV4SWS.KOM and LOG4SWS.KOM differ in their treatment of semantic annotations, they are based on the identical matchmaker framework. As a result, they share a significant number of characteristics.

Most importantly, both matchmakers employ the notion of *operations-focused matching*. An overview of the process is depicted in Figure 1.1. In detail, individual similarity values are computed for the components on all levels of abstraction in a service, i.e., interfaces, operations, inputs, and outputs (yielding sim_{iface}, sim_{op},

---

[1] The names of our matchmakers have historical roots: COV was traditionally based on the determination of the degree of coverage between semantic concepts; LOG refers to logic subsumption matching. The common name component 4SWS means "for Semantic Web Services", KOM refers to the abbreviated name of our institute at Technische Universitt Darmstadt.

[2] As a matter of fact, both matchmakers are also applicable to service description formalisms that exhibit a structure similar to (SA)WSDL. An application of LOG4SWS.KOM to hRESTS with MicroWSMO annotations – service description formalisms for RESTful services – has been presented by Lampe et al. [12]

sim_{in}, and *sim_{out}* respectively). Subsequently, these similarity values are aggregated using a linear function on the level of operations, resulting in *sim_{agg}*. For this aggregation process, the respective weights may be freely chosen ($w_{iface}, w_{op}, w_{in}, w_{out}$). The weights thus account for varying expressiveness of (semantic) descriptions on the different service abstraction levels.

Subsequently, the objective of the matching process consists in the determination of an optimal pairing between operations in a service request and service offer. This procedure is based on the notion that operations constitute *the* essential unit of functionality in a service. Based on the similarities of the paired operations, the *average similarity* of both services is determined.

The similar notion of average similarity is, for instance, applied by Plebani and Percini in the *URBE* matchmaker [22]. This differs from the concept of *global DoMs*, as initially defined by Paolucci et al. [21], further elaborated by Bellur and Kulkarni [1], and, for instance, applied by Klusch et al. in the *SAWSDL-MX* matchmaker [8]. A global DoM can be interpreted as minimal lower bound of similarity a service offer guarantees with respect to a given query. In contrast, the average similarity can be interpreted as the amount of effort that is required to adapt a service offer to the requirements of the service consumer. Both approaches have their pros and cons; a more extensive discussion can be found in our previous work on LOG4SWS.KOM [27].

Another common aspect in both matchmakers is the utilization of a fallback strategy based on the WordNet ontology [18]. It comes into effect if semantic annotations are unavailable or cannot be processed for a certain service component. In this case, the similarity value is determined based on the inverse distance between the individual English words in a component name. It is important to note that the fallback-strategy only serves as a *substitute*, rather than a *complement* to the use of semantic information. That is, for services that are fully semantically annotated, our matchmakers exclusively rely on semantic information.

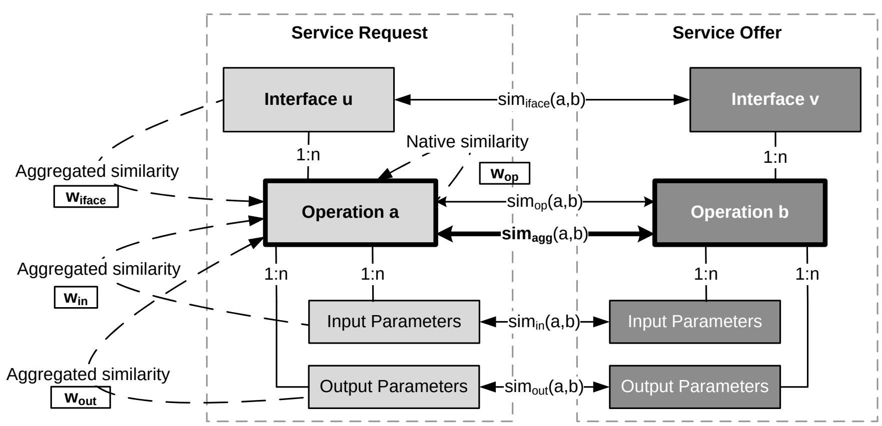

Fig. 1.1 Overview of the matchmaking process

1.2.2 LOG4SWS.KOM: Adapted Subsumption Matching

Matchmaking based on traditional subsumption reasoning often relies on *discrete* DoMs. For instance, in their seminal paper on semantic matchmaking, Paolucci et al. [21] propose the DoMs *exact*, *plug in*, *subsume*, and *fail*. Subsumption matching enjoys substantial popularity in the domain of semantic matchmaking and is, e.g., applied by Klusch et al. [8], Cardoso [4], and Li and Horrocks [13].

However, the utilization of discrete DoMs can be associated with three major disadvantages. First, it results in a relatively coarse-grained ranking of services; a subsequent, more fine-grained ranking requires the inclusion of additional (generally non-logic) similarity assessments. Second, the approach complicates or inhibits the combination with additional similarity measures, such as word similarity, if a hybrid matching methodology is applied. Third, the approach makes basic assumptions regarding the generalization and specialization of semantic concepts in ontologies which are not necessarily met. A more elaborate discussion of this issue can be found in our previous work [27].

LOG4SWS.KOM addresses these shortcomings by mapping discrete DoMs onto a continuous numerical scale. Such mapping procedure has been proposed in the past [15, 5]. However, the determination of correspondences between DoMs and numerical equivalents is fairly arbitrary in nature. LOG4SWS.KOM avoids this problem by using an *Ordinary Least Squares* (OLS) estimator. The estimator automatically determines numerical equivalents for each DoM for each service abstraction level and thus self-adapts to a given training data set. As a more fine-grained complement to the subsumption DoM, the inverse minimal path length between semantic concepts is additionally taken into consideration. The principal concept of such measure can be traced back to the *edge counting* approaches that were first introduced by Rada et al. [24].

For detailed information on all aspects of LOG4SWS.KOM, please refer to our previous publication [27].

1.2.3 COV4SWS.KOM: Semantic Relatedness

Through its mapping mechanism, LOG4SWS.KOM alleviates the problem of a coarse-grained ranking of service. However, the similarity assessment is ultimately still based on logic subsumption matching with discrete DoMs. Thus, in COV4SWS.KOM, we follow an alternative approach. Namely, we apply similarity measures from the field of semantic relatedness.

The assignment of semantic relatedness of concepts in an ontology or taxonomy is a well-known problem from computational linguistics and artificial intelligence. Research in this domain has resulted in a variety of different similarity measures; for additional details, we refer to a survey by Budanitsky and Hirst [3].

For COV4SWS.KOM, we have selected the measures by Lin [14] and Resnik [25]. Both approaches require the assignment of probabilities of occurrence to all semantic concepts in the utilized ontologies. Due to the lack of a standard corpus, we exploit the set of registered services in our matchmaker as training corpus.

COV4SWS.KOM self-adapts to a given set of training data through the automatic determination of service level weights. For that purpose, an OLS estimator is used. Thus, COV4SWS.KOM can automatically account for fluctuating richness and expressiveness of semantic and syntactic information on different abstraction levels of a service.

1.3 Solution

COV4SWS.KOM and LOG4SWS.KOM have been implemented in Java based on the identical technical foundation. We apply Pellet as reasoner[3] and Java WordNet Libary (JWNL)[4] as interface to WordNet. We further utilize a proprietary parser, based on the Java Document Object Model (JDOM) framework[5], in order to process (SA)WSDL files.

A number of restrictions apply for both matchmakers: In the parsing process, we solely regard the topmost level of parameters, i.e., only those XML Schema *types* or *elements* that are directly referred to by a WSDL message. With respect to semantic annotations, we make exclusive use of *modelReference* attributes (i.e., *schemaMappings* are not taken into account). As an additional restriction, only the first semantic concept (i.e., the first value of modelReference) for each component is utilized in the matchmaking process. In our experience, aforementioned restrictions are common in the semantic matchmaking domain.

In order to compute an optimal matching of components (specifically, inputs, outputs, and operations), bipartite graphs are employed, as initially suggested by Guo et al. [6]. For this purpose, an implementation of the *Hungarian algorithm* by Nedas is applied [20]. This implementation also supports bipartite graphs with differing cardinalities in the two partitions. The principal extension of the Hungarian algorithm for this purpose has been suggested by Bourgeois and Lassalle [2].

For the OLS estimator, input data is acquired by matching a given set of example service requests against a given set of example offers. In case of LOG4SWS.KOM, the so-called design matrix [23] – commonly abbreviated X – is given by the relative frequency of the four DoMs (weighted using the inverse path length) on all levels of abstraction in a service. In case of COV4SWS.KOM, the design matrix is inferred from the similarity values on all service levels. For both matchmakers, the vector of predictors – commonly abbreviated y – is inferred from a given binary or graded relevance rating. Each pair of service request and offer yields one row in the design matrix and vector of predictors respectively. For further details, we refer the interested reader to our previous work [27].

[3] http://clarkparsia.com/pellet/

[4] http://jwordnet.sourceforge.net/

[5] http://www.jdom.org/

In order to speed up the matching process, we utilize various caching mechanisms. In detail, this includes a cache for splitting component names into individual English words, a cache for word distances in WordNet, and a cache of subsumption relations and path lengths between semantic concepts. Caches may be populated at both registration- and query-time. We additionally map all service descriptions into a lightweight internal model. This model essentially only holds the names and semantic concepts for all service components, as well as their parent-child relations.

1.4 Lessons Learned

1.4.1 Evaluation Results

For both matchmakers we have evaluated multiple configurations, i.e., variants and corresponding versions.

For COV4SWS.KOM, the variation concerns the level weights (variants) and the applied similarity metrics (versions). Variant 1 is signature-based, i.e., we assign a level weight of 0.5 to both inputs and outputs and a level weight of 0 to operations and interfaces. Furthermore, the fallback-strategy is disabled in this variant. In Variant 2, we utilize weights of 0.4 for inputs and outputs and 0.1 for operations and interfaces. This weighting accounts for the fact that only the parameter level is annotated in our utilized test collections. In Variant 3, we follow a naive approach and assign equal weights of 0.25 to all service abstraction levels. In Variant 4, we utilize OLS for the determination of optimal weights. Version A makes use of Lin's similarity measure, whereas Version 2 applies Resnik's measure.

For LOG4SWS.KOM, the variation concerns the level weights (variants) and the numerical DoM mappings (versions). We only evaluate Variants 1 through 3, with the identical definition as for COV4SWS.KOM. For Version A, the numerical equivalents have been set to 1 for exact matches and 0 for fail matches. Plug-in and subsume matches are assigned an numerical equivalent of 0.5. This reflects their intermediate positions between exact and fail matches. The same idea is followed by Syeda-Mahmood [28]. In Version B, we apply OLS for the determination of numerical DoM equivalents.

An overview of all configurations of COV4SWS.KOM and LOG4SWS.KOM is provided in Tables 1.1 and 1.2 respectively. Level weights are given in the order interface, operation, input, output; numerical DoM equivalents are in the order exact, plug-in, subsume, and fail[6]. Configuration 4B of COV4SWS.KOM and configuration 2B of LOG4SWS.KOM are marked with "S3" because they participated in the S3 Contest in 2010 [10].

[6] In fact, we utilize generic definitions of DoMs in LOG4SWS.KOM that slightly deviates from the ones introduced by Paolucci et al. [21]. However, this does not have any practical implications for the evaluated configurations. For additional details, please refer to our previous work [27].

Table 1.1 Evaluated Configurations of COV4SWS.KOM

Config.	Level Weights	Sim. Measure
1A	0, 0, 0.5, 0.5	Lin
1B	0, 0, 0.5, 0.5	Resnik
2A	0.1, 0.1, 0.4, 0.4	Lin
2B	0.1, 0.1, 0.4, 0.4	Resnik
3A	0.25, 0.25, 0.25, 0.25	Lin
3B	0.25, 0.25, 0.25, 0.25	Resnik
4A	From OLS	Lin
4B (S3)	From OLS	Resnik

Table 1.2 Evaluated Configurations of LOG4SWS.KOM

Config.	Level Weights	Num. DOM Equivalents
1A	0, 0, 0.5, 0.5	1, 0.5, 0.5, 0
1B	0, 0, 0.5, 0.5	From OLS
2A	0.1, 0.1, 0.4, 0.4	1, 0.5, 0.5, 0
2B (S3)	0.1, 0.1, 0.4, 0.4	From OLS
3A	0.25, 0.25, 0.25, 0.25	1, 0.5, 0.5, 0
3B	0.25, 0.25, 0.25, 0.25	From OLS

For all aforementioned configurations, we conducted separate evaluation runs using SAWSDL-TC1 (which constituted the basis for the SAWSDL track of the S3 Contest in 2009) and SAWSDL-TC3 (S3 Contest in 2010)[7].

For the configurations where OLS was applied, the full test collections served as training data. In the evaluation, we apply *k*-fold cross-validation [19]. In the example at hand, *k* corresponds to the number of queries in the utilized test collection (i.e., 26 for SAWSDL-TC1 and 42 for SAWSDL-TC3), because every query and corresponding relevance set from SAWSDL-TC1 and TC3 respectively serves as a partition from the service set. That is, in the evaluation, for each individual query, all data sets that refer to this query are eliminated from OLS training data.

Even though SAWSDL-TC3 contains an additional graded relevance rating, we made exclusive use of the binary relevance rating for training purposes. This procedure allows for a better comparison of results. For the actual evaluation process, the *Semantic Matchmaker Evaluation Environment* (SME2) was applied[8].

A summary of evaluation results is provided in Tables 1.3 through 1.6. For each configuration, we include the *Information Retrieval* (IR) metrics that SME2 automatically computes, namely *Average Precision* (AP), *Q-measure* (Q), and *normalized Discounted Cumulative Gain* (nDCG). Apostrophes (') denote the adapted metrics for incomplete relevance sets. Furthermore, *Precision at 5* (P5), *Precision at 10* (P10), and *r-Precision* (RP) are provided in the result overview. The best respective value for each metric is highlighted in boldface.

In the case of SAWSDL-TC1, the metrics based on graded relevance (Q, Q', *nDCG*, and *nDCG*) have been omitted because they cannot be computed due to the

[7] Both test collections are available at http://projects.semwebcentral.org/projects/sawsdl-tc/

[8] http://projects.semwebcentral.org/projects/sme2/

lack of predefined ratings. For an overview and formal definition of aforementioned metrics, we refer to Sakai and Kando [26] and Manning et al. [16] .

We refrain from the inclusion of *Query Response Time* (QRT) results. In our opinion, the characteristics of the machine that is used for evaluation and the utilization of caches renders concrete QRT figures largely incomparable. The interested reader is, however, referred to the results of the 2010 S3 Contest for QRT rankings [10]. We further provide a qualitative discussion of the matter in Section 1.4.2.

Table 1.3 Evaluation Results for COV4SWS.KOM and SAWSDL-TC1

Config.	AP	AP'	Q	Q'	nDCG	nDCG'	P5	P10	RP
1A	0.644	0.644	–	–	–	–	0.731	0.727	0.603
1B	0.665	0.665	–	–	–	–	0.723	0.742	0.605
2A	0.737	0.737	–	–	–	–	0.915	0.835	0.672
2B	0.752	0.752	–	–	–	–	0.885	0.819	0.693
3A	0.743	0.743	–	–	–	–	**0.954**	0.846	0.684
3B	**0.778**	**0.778**	–	–	–	–	0.931	0.827	**0.707**
4A	0.722	0.722	–	–	–	–	0.892	0.831	0.669
4B	0.755	0.755	–	–	–	–	0.908	**0.850**	0.681

Table 1.4 Evaluation Results for COV4SWS.KOM and SAWSDL-TC3

Config.	AP	AP'	Q	Q'	nDCG	nDCG'	P5	P10	RP
1A	0.617	0.710	0.616	0.725	0.692	0.787	0.724	0.681	0.599
1B	0.635	0.734	0.598	0.708	0.662	0.760	0.686	0.674	0.615
2A	0.706	0.784	0.728	0.806	0.809	0.873	0.795	0.738	0.655
2B	0.710	0.796	0.706	0.791	0.780	0.851	0.776	0.712	0.658
3A	0.710	0.796	0.729	0.812	0.803	0.867	**0.800**	0.733	0.661
3B	0.726	0.808	0.727	0.808	0.803	0.869	0.790	0.726	0.683
4A	0.680	0.802	0.701	0.813	0.791	0.877	0.795	0.733	0.624
4B	**0.736**	**0.823**	**0.741**	**0.825**	**0.818**	**0.884**	0.790	**0.755**	**0.686**

Table 1.5 Evaluation Results for LOG4SWS.KOM and SAWSDL-TC1

Config.	AP	AP'	Q	Q'	nDCG	nDCG'	P5	P10	RP
1A	0.718	0.718	–	–	–	–	0.869	0.792	0.647
1B	0.742	0.742	–	–	–	–	0.846	0.827	0.678
2A	0.747	0.747	–	–	–	–	0.931	0.842	0.685
2B	**0.808**	**0.808**	–	–	–	–	**0.962**	**0.885**	**0.735**
3A	0.725	0.725	–	–	–	–	0.931	0.842	0.661
3B	0.758	0.758	–	–	–	–	0.954	0.835	0.699

Table 1.6 Evaluation Results for LOG4SWS.KOM and SAWSDL-TC3

Config.	AP	AP'	Q	Q'	nDCG	nDCG'	P5	P10	RP
1A	0.666	0.750	0.667	0.768	0.741	0.824	0.757	0.721	0.617
1B	0.690	0.785	0.692	0.795	0.757	0.846	0.743	0.710	0.648
2A	0.720	0.797	0.744	0.820	0.815	0.877	0.795	**0.764**	0.651
2B	**0.763**	**0.837**	**0.778**	**0.851**	**0.836**	**0.896**	**0.800**	0.755	**0.709**
3A	0.696	0.792	0.725	0.813	0.797	0.875	0.795	**0.764**	0.623
3B	0.721	0.814	0.745	0.831	0.810	0.882	0.795	**0.764**	0.653

1.4.2 Advantages and Disadvantages

As can be seen from the evaluation results, both matchmakers generally perform worse for SAWSDL-TC3 in comparison to SAWSDL-TC1 with respect to *AP*. The difference amounts to roughly 0.05 points across all versions and variants. In the following, if not noted differently, our discussion will concern the evaluation results for SAWSDL-TC3. However, the findings are also valid for SAWSDL-TC1.

In general, with the results of the S3 Contest in 2010 serving as a basis, both matchmakers deliver a very competitive matchmaking performance for most configurations. LOG4SWS.KOM performs slightly better than COV4SWS.KOM with respect to all considered metrics[9]. In the following, we will discuss both matchmakers separately. Subsequently, we provide a discussion of common observations and additional findings from the development process.

The results for COV4SWS.KOM indicate that metrics from the field of semantic relatedness are well applicable to the problem of matchmaking. Variants B, which utilize Resnik's similarity measure outperform Variants A, which are based on Lin's metric, at all level weights in terms of *AP*, *P*5, *P*10, and *RP*. Interestingly, for the evaluation metrics that are based on graded relevance (namely, Q, Q', $nDCG$, and $nDCG'$), the picture is more or less reversed.

A potential explanation may lie in the elementary difference between Lin's and Resnik's measure. Whereas Lin's measure is normalized to a value range of $[0,1]$, Resnik's measure may correspond to $[0,\infty[$. This has two implications: First, because the WordNet-based fallback strategy also provides value in the range $[0,1]$, Resnik's measure (and thus, the semantic information) is potentially overweighted in the aggregation process. Second, the releative difference between similarity values for similar and non-similar semantic concepts may be larger with Resnik's measure, due to the unbounded value range. A speculation is that this leads to two "clusters" of rather small and rather large similarity values. These value partitions may be very good determinants of binary relevance (which corresponds to two clusters of *relevant* and *non-relevant* services).

As the comparison of Version 1 with Versions 2 and 3 shows, the matchmaking performance of COV4SWS.KOM notably profits from the inclusion of service

[9] Using the mean average of each metric across the comparable Variants 1A through 3B as a basis for comparison.

abstraction levels beyond the service signature (i.e., inputs and outputs). In fact, there also is a small improvement in all metrics observable between Version 2 and Version 3. This indicates that syntatic information (i.e., the names of interface and operation components) can be of similar importance as semantic information in the determination of service similarity, at least as far as SAWSDL-TC1 and -TC3 are concerned.

This assumption is supported by a manual examination of relevance sets in the SAWSDL-TC3. For most services within each relevance set, the names of both interface and operation components are either very similar or identical. In this case, our rather rudimentary fallback-strategy correctly determines very high similarity values. However, the approach is vulnerable to false positives in cases where the lexical similarity between two distinct relevance sets is high. An example are the *bookpersoncreditcardaccount__service.wsdl* and *bookpersoncreditcardaccount_price_service.wsdl* queries and corresponding relevance sets in the SAWSDL-TC3. In these specific cases of high lexical service similarity, the semantic concepts associated with inputs and outputs can provide significantly more discriminative power.

With respect to Version 4, where OLS is applied, we observe a diametric effect on Variants A and B. For Variant A, i.e., the variant based on Lin's measure, the utilization of OLS results in a deterioration of matchmaking performance in terms of most metrics, as compared to the manual level weights in Versions 2 and 3. It is worthy to note that Klusch et al. [8] have made a similar observation – namely, the degradation of matchmaking performance due to the use of machine-learning techniques – in their SAWSDL-MX2 matchmaker.

In contrast, for Variant B (based on Resnik's measure), we observe a notable increase in most metrics with OLS, as compared to the manually configured variants. In fact, configuration 4B delivers the best overall matchmaking performance of all configurations of COV4SWS.KOM. Again, this may potentially be attributed to the characteristics of Resnik's measure, which have been previously discussed.

As the evaluation results for LOG4SWS.KOM show, our adapted variant of subsumption reasoning provides very promising matchmaking results. In fact, configuration 2B of LOG4SWS.KOM achieved the first place in the S3 Contest in 2010 concerning the *Q* and *nDCG* metrics. It only trailed *iSEM*, a matchmaker by Klusch et al. [7], by a small margin regarding *AP*. This indicates that improvements in semantic matchmaking performance do not necessarily require revolutionary changes; in fact, our extension of traditional and well-proven subsumption matching with an OLS estimator is rather evolutionary in nature.

In line with our observations for COV4SWS.KOM, the hybrid Versions 2 and 3 of LOG4SWS.KOM perform significantly better than the signature-based Version 1. The improvement is observable for all regarded evaluation metrics. However, with respect to most evaluation metrics (i.e, *AP*, *Q*, and *nDCG*), Version 3 – which puts higher weight on the not semantically annotated interface and operations levels – performs worse in comparison to Version 2. This implies that the determination of semantic similarity works very well in LOG4SWS.KOM with respect to the overall task of determining service similarity. Accordingly, the semantically annotated input and output levels should be assigned greater weight.

In this context, the utilization of path length as a complimentary measure of similarity appears to be a good choice for the SAWSDL-TC1 and -TC3. In fact, URBE – the matchmaker which achieved the highest *AP* in the 2009 S3 Contest's SAWSDL track [9] – uses the path length as the exclusive measure of semantic similarity [22]. The methodology we apply in LOG4SWS.KOM can be interpreted as a weighted path length measure, with the weights depending on the basic subsumption type.

Across all versions, Variants B, which utilize OLS, deliver an improved matchmaking performance compared to the manually configured Variants A. The effect is most notable for the AP and AP' metrics in terms of absolute gain. This comes as little surprise, because the training data is based on the binary relevance rating of services. Accordingly, the improvement should be the highest for those metrics that are computed based on binary relevance, namely AP and AP'. In practice, however, the use of the more fine-grained graded relevance is preferable for the training process, because it likely leads to a better overall fit of the OLS estimation.

A common disadvantage of COV4SWS.KOM and LOG4SWS.KOM is the need for suitable training data if the self-adaptation mechanisms are to be exploited. Suitability, in this context, refers to the following minimal requirements: The training data has to comprise a set of service queries and a set of service offers, and at least a part of the resulting query/offer-pairs has to be associated with some form of numerical relevance ranking. As the example of the SAWSDL-TC1 and -TC3 demonstrates, the process of assigning rankings commonly requires significant human effort. Additionally, the training data should be representative of the services that are generally processed by the matchmaker. If the latter condition is not met, the matchmaker performance may strongly deteriorate. This regard, it should be noted that the utilization of the whole test collection in our evaluation constitutes an ideal scenario that will commonly not occur in practice.

A similar problem is the need for a representative corpus of semantic concepts. As outlined in Section 1.2.3, the application of Lin's and Resnik's measure requires the assignment of probabilities of occurrence to all referenced semantic concepts. In case of the English language, for instance, the so-called *Brown Corpus* [11] provides a well-established source of information for this purpose; a likewise corpus for SWS would be desirable. Again, the utilization of all registered services in our matchmaker can only be considered a temporary solution. Specifically, in practical application, the registration of new service offers and the processing of queries will occur intermittedly. Thus, following each registration of a service that contains previously unknown semantic concepts, all probabilities would have to be reassigned.

An additional drawback consists in the computational effort of the training phase. In detail, each service query in the training set has to be matched against each offer. In the case of SAWSDL-TC3 with its 42 queries and 1,080 offers, for example, this results in a total of 45,360 matchmaking operations. Even under the assumption that each matchmaking operation solely requires 100 milliseconds, the training phase would take approximately 75 minutes. Whether this is problematic in practice largely depends on size of the training data and the frequency at which the training phase is repeated.

On the positive side, the computational effort for the OLS estimator is relatively low. In fact, for the utilized test collections with approximately 1,000 service offers, the determination of level weights (in COV4SWS.KOM) or numerical DoM equivalents (LOG4SWS.KOM) takes less than 100 milliseconds per query. This figure includes the required time for the preliminary filtering and partitioning of the input data, which is triggered by the application of cross-validation, cf. Section 1.4.1.

In this context, it is interesting to note that the fallback strategy-related operations require the most computational effort in our matchmakers. URBE, for instance, implements a rather simple strategy for splitting component names into words, based on common separators, such as dash (-) or underscore (_). Such a strategy can very efficiently be implemented using a string tokenizer. However, it fails for names such as *Academic-degreegovernmentorganizationFundingSoap*, which occur in the SAWSDL-TC1 and -TC3. In our matchmakers, we employ a strategy that recursively extracts substrings from component names and checks for the existence of these substrings in the WordNet ontology. For the given example name, this results in dozens of lookup operations. In contrast, the comparison of two semantic concepts via subsumption reasoning is far less "costly" in terms of required computational effort.

Generally, the matchmaking process significantly profits from the utilization of caches. In fact, query response times of a few hundred milliseconds can only be realized through efficient caching mechanisms. In the optimal case where all required similarity assessments are cached, only the computation of component assignments is required. Using the Hungarian algorithm, this process can be conducted in a few milliseconds for each pair of services. In this respect, both COV4SWS.KOM and LOG4SWS.KOM profit from their preliminary training phase, because it leads to an optimal population of caches for the already processed pairs of queries and offers.

Lastly, we would like to discuss an useful observation that is not evident from the presented evaluation results, but was made throughout the development process of our matchmakers. As discussed in section 1.3, our approach considers only the topmost level of XML schema declarations in the matchmaking process.. We consider this approach valid, because the underlying parameter structure seems to have been introduced through the semi-automatic conversion process from OWL-S TC[10] to SAWSDL-TC. Thus, according to our experience, the XSD structure beneath the topmost types does not have any informational value as far as the SAWSDL-TC is concerned. In fact, as the evaluation results for an initial implementation of the WSDL parser showed, the matchmaking performance degraded if the structure of each *complexType* was parsed. This degradation concerned both the runtime as well as the precision of the matchmaking process.

[10] http://www.semwebcentral.org/projects/owls-tc/

1.4.3 Conclusions and Future Work

From the evaluation results presented in this chapter, a wide range of conclusions can be drawn.

As the results for COV4SWS.KOM indicate, similarity metrics from the field of semantic relatedness can be applied to the problem of service matchmaking with promising results. These metrics also have one significant advantage over traditional subsumption matching: They natively provide a numerical similarity assessment on a continuous scale. Thus, metrics of semantic relatedness can immediately be integrated with other similarity measures. This is specifically helpful in a (hybrid) matchmaking process with weighted similarity aggregation, such as implemented in our matchmakers. A possible future extension concerns the integration of additional similarity measures; for that matter, a survey by Budanitsky and Hirst [3] may provide a good starting point.

The results of LOG4SWS.KOM show that an extension of classical subsumption matching through a numerical mapping process may also yield very competitive matchmaking performance. In this case, the arbitrary nature of such a mapping process can be addressed through the inclusion of a self-adaptation mechanism; in our case, an OLS estimator. A possible future revision of LOG4SWS.KOM concerns the extension of the OLS estimator to the determination of level weights, as it is applied in COV4SWS.KOM. However, this would also require a more extensive two-phase training process, because the determination of optimal level weights depends on a prior determination of numerical DoM equivalents.

In general, the application of self-adaptation mechanisms provides improved results compared to a manual configuration. This is true for both matchmakers, COV4SWS.KOM and LOG4SWS.KOM. However, in line with previous findings by Klusch et al. [8], we also observed a deterioration in matchmaking performance for selected variants. A straightforward extension in future work concerns the integration of additional adaptation mechanisms. In fact, the domain of machine learning provides a rich source of applicable techniques for that matter, cf. Witten and Frank [29].

An observation for both matchmakers is that hybrid matchmaking and the inclusion of additional service abstraction levels has a positive impact on matchmaking performance. Unfortunately, the effects cannot be quantified for both features individually, because neither SAWSDL-TC1 nor -TC3 provides semantic annotations on the interface or operation level. Thus, the syntax-based fallback strategy (and only the fallback strategy) comes into effect on these abstraction levels. However, it is interesting to observe the increase in matchmaking performance despite the fact that a rather rudimentary matchmaking strategy – namely, inverse WordNet distance – is applied in our matchmakers. A potential future extension consists in the integration of more sophisticated fallback strategies. In fact, the same methods from the field of semantic relatedness that we already apply in COV4SWS.KOM are candidates of interest in this respect, because they originate in the area of language processing.

1.5 Summary

In the chapter at hand, we presented COV4SWS.KOM and LOG4SWS.KOM, two self-adaptive matchmakers for semantic Web services that operate on the WSDL description format with SAWSDL annotations. Both matchmakers are based on the identical platform, XAM4SWS, and thus share a large amount of common features.

Most notably, this includes an operations-focused matchmaking approach that aggregates similarities of different service abstraction levels on the level of operations. Additional common aspects are the determination of average service similarity, reflecting the required effort for adaption to service consumer demands. Lastly, a WordNet-based fallback strategy is employed in both matchmakers.

COV4SWS.KOM uses methods from the field of semantic relatedness – namely the metrics by Lin and Resnik – for the computation of similarity between semantic concepts. The matchmaker further utilizes an OLS estimator to determine optimal weights for the aggregation of similarity values from different abstraction levels.

LOG4SWS.KOM makes use of traditional subsumption matching, but maps the resulting discrete DoMs to numerical equivalents. For this mapping process, an OLS estimator may be utilized. The inverse path length serves as complementary similarity measure.

As our evaluation on the basis of SAWSDL-TC1 and -TC3 shows, both matchmakers provide very competitive results in terms of common IR metrics. Specifically, the results for COV4SWS.KOM indicate the principal applicability of metrics from the field of semantic relatedness to the problem of SWS matchmaking. At the same time, the evaluation LOG4SWS.KOM leads us to conclude that the combination of two rather "traditional" matchmaking approaches (namely, subsumption reasoning and path length measure) may also be very efficient.

While LOG4SWS.KOM significantly profits from its self-adaptation mechanism, we obtain mixed results for COV4SWS.KOM. In fact, for selected variants, a notable deterioration in matchmaking performance can be observed. Further, the selection of a representative service set for training may constitute a challenge in practical application.

Both matchmakers heavily profit from the inclusion of (not semantically annotated) service abstraction levels beyond the service signature in the matchmaking process. This is true in spite of the rather rudimentary nature of our implemented fallback strategy, which is based on the inverse WordNet distance.

In our future work, we will primarily focus on two points. The first concerns the inclusion of additional similarity metrics from the field of semantic relatedness in COV4SWS.KOM. In fact, this domain offers a wide range of well-explored methodologies that could be adapted to the problem of semantic matchmaking with comparatively little effort. The previously mentioned survey paper by Budanitsky and Hirst [3], for instance, includes a comparative assessment of five similarity measures (including those by Lin and Resnik), both information- and path-length-based. In addition, these measures may also act as a substitute or extension to the rather rudimentary, WordNet-based fallback strategy that we have implemented so far.

The second point concerns the implementation of additional self-adaptation mechanisms. In this area, research in IR and data mining provides a rich set of options. In fact, the popular Weka toolkit by Witten and Frank [29] implements a multitude of different machine learning techniques that are potentially suited for the purpose of semantic matchmaking, such as decision trees or support vector machines.

Final Note

In order to permit an independent assessment and verification of the evaluation results for COV4SWS.KOM and LOG4SWS.KOM through the SWS research community, the complete XAM4SWS matchmaker framework is available via SemWebCentral at http://projects.semwebcentral.org/projects/xam4sws.

References

1. Bellur, U., Kulkarni, R.: Improved Matchmaking Algorithm for Semantic Web Services Based on Bipartite Graph Matching. In: 2007 IEEE International Conference on Web Services, pp. 86–93 (2007)
2. Bourgeois, F., Lassalle, J.C.: An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices. Communications of the ACM **14**(12), 802–804 (1971)
3. Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics **32**(1), 13–47 (2006)
4. Cardoso, J.: Discovering Semantic Web Services with and without a Common Ontology Commitment. In: Third International Semantic and Dynamic Web Processes Workshop, pp. 183–190 (2006)
5. Fernández, A., Polleres, A., Ossowski, S.: Towards Fine-grained Service Matchmaking by Using Concept Similarity. In: First International Joint Workshop SMR^2 2007 on Service Matchmaking and Resource Retrieval in the Semantic Web at the 6th International Semantic Web Conference, vol. 243, pp. 31–46 (2007)
6. Guo, R., Chen, D., Le, J.: Matching Semantic Web Services across Heterogeneous Ontologies. In: Fifth International Conference on Computer and Information Technology, pp. 264–268 (2005)
7. Klusch, M., Kapahnke, P.: iSeM: Approximated Reasoning for Adaptive Hybrid Selection of Semantic Services. The Semantic Web: Research and Applications **6089**, 30–44 (2010)
8. Klusch, M., Kapahnke, P., Zinnikus, I.: Hybrid Adaptive Web Service Selection with SAWSDL-MX and WSDL-Analyzer. In: L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvnen, R. Mizoguchi, E. Oren, M. Sabou, E. Simperl (eds.) The Semantic Web: Research and Applications, *Lecture Notes in Computer Science*, vol. 5554, pp. 550–564. Springer (2009)
9. Klusch, M., Leger, A., Martin, D., Paolucci, M., Bernstein, A., Küster, U.: 3rd International Semantic Service Selection Contest – Retrieval Performance Evaluation of Matchmakers for Semantic Web Services (S3 Contest). In: Third International Workshop SMR^2 2009 on Service Matchmaking and Resource Retrieval in the Semantic Web at the 8th International Semantic Web Conference (2009)
10. Klusch, M., Leger, A., Martin, D., Paolucci, M., Bernstein, A., Küster, U.: 4rd International Semantic Service Selection Contest – Retrieval Performance Evaluation of Matchmakers for

Semantic Web Services (S3 Contest). In: Fourth International Workshop SMR^2 2009 on Service Matchmaking and Resource Retrieval in the Semantic Web at the 9th International Semantic Web Conference (2010)
11. Kucera, H., Francis, W.N.: Computational Analysis of Present-Day American English. Brown University Press (1967)
12. Lampe, U., Schulte, S., Siebenhaar, M., Schuller, D., Steinmetz, R.: Adaptive Matchmaking for RESTful Services based on hRESTS and MicroWSMO. In: 5th Workshop on Enhanced Web Service Technologies, pp. 10–17 (2010)
13. Li, L., Horrocks, I.: A Software Framework for Matchmaking Based on Semantic Web Technology. International Journal of Electronic Commerce **8**(4), 39–60 (2004)
14. Lin, D.: An Information-Theoretic Definition of Similarity. In: Fifteenth International Conference on Machine Learning, pp. 296–304 (1998)
15. Liu, C., Peng, Y., Chen, J.: Web Services Description Ontology-Based Service Discovery Model. In: 2006 IEEE / WIC / ACM International Conference on Web Intelligence, pp. 633–636 (2006)
16. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
17. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic Web Services. IEEE Intelligent Systems **16**(2), 46–53 (2001)
18. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM **38**(11), 39–41 (1995)
19. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
20. Nedas, K.A.: Munkres' (Hungarian) Algorithm. Available online at http://konstantinosnedas.com/dev/soft/munkres.htm, last access on February 21, 2011 (2005)
21. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Importing the Semantic Web in UDDI. In: International Workshop on Web Services, E-Business, and the Semantic Web in Connection with the 14th Conference on Advanced Information Systems Engineering, pp. 225–236 (2002)
22. Plebani, P., Pernici, B.: URBE: Web Service Retrieval Based on Similarity Evaluation. IEEE Transactions on Knowledge and Data Engineering **21**(11), 1629–1642 (2009)
23. Rachev, S.T., Mittnik, S., Fabozzi, F.J., Focardi, M., Jasic, T.: Financial Econometrics: From Basics to Advanced Modeling Techniques. John Wiley & Sons (2007)
24. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and Application of a Metric on Semantic Nets. IEEE Transactions on Systems, Man and Cybernetics **19**(1), 17–30 (1989)
25. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Fourteenth International Joint Conference on Artificial Intelligence, pp. 448–453 (1995)
26. Sakai, T., Kando, N.: On Information Retrieval Metrics Designed for Evaluation with Incomplete Relevance Assessments. Information Retrieval **11**(5), 447–470 (2008)
27. Schulte, S., Lampe, U., Eckert, J., Steinmetz, R.: LOG4SWS.KOM: Self-Adapting Semantic Web Service Discovery for SAWSDL. In: Fourth International Workshop of Software Engineering for Adaptive Service-Oriented Systems (2010)
28. Syeda-Mahmood, T., Shah, G., Akkiraju, R., Ivan, A.A., Goodwin, R.: Searching Service Repositories by Combining Semantic and Ontological Matching. In: 2005 IEEE International Conference on Web Services, pp. 13–20 (2005)
29. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, second edn. Morgan Kaufmann (2005)
30. Wooldridge, J.: Introductory Econometrics: A Modern Approach, 4 edn. South-Western Cengage Learning (2008)