A Session-based Approach for Aligning Large Ontologies

Patrick Lambrix[1,2] and Rajaram Kaliyaperumal[1]

(1) Department of Computer and Information Science
(2) Swedish e-Science Research Centre
Linköping University, 581 83 Linköping, Sweden

Abstract. There are a number of challenges that need to be addressed when aligning large ontologies. Previous work has pointed out scalability and efficiency of matching techniques, matching with background knowledge, support for matcher selection, combination and tuning, and user involvement as major requirements. In this paper we address these challenges. Our first contribution is an ontology alignment framework that enables solutions to each of the challenges. This is achieved by introducing different kinds of interruptable sessions. The framework allows partial computations for generating mapping suggestions, partial validations of mappings suggestions and use of validation decisions in (re)computation of mapping suggestions and the recommendation of alignment strategies to use. Further, we describe an implemented system providing solutions to each of the challenges and show through experiments the advantages of the session-based approach.

This is a preprint of the paper published by Springer:
Lambrix P, Kaliyaperumal R, A Session-based Approach for Aligning Large Ontologies, Tenth Extended Semantic Web Conference - ESWC 2013, LNCS 7882, 46-60, Montpellier, France, 2013.
The final publication is available at www.springerlink.com.
doi: 10.1007/978-3-642-38288-8

1 Introduction

In recent years many ontologies have been developed and many of those contain overlapping information. Often we want to use multiple ontologies. For instance, companies may want to use community standard ontologies and use them together with company-specific ontologies. Applications may need to use ontologies from different areas or from different views on one area. In each of these cases it is important to know the relationships between the terms in the different ontologies. Further, the data in different data sources in the same domain may have been annotated with different but similar ontologies. Knowledge of the inter-ontology relationships would in this case lead to improvements in search, integration and analysis of data. It has been realized that this is a major issue and much research has recently been done on ontology alignment, i.e. finding mappings between terms in different ontologies (e.g. [5]).

The existing frameworks for ontology alignment systems (e.g. [3, 13]) describe different components and steps in the ontology alignment process such as preprocessing, matching, filtering and combining match results, and user validation of the mapping suggestions generated by the ontology alignment system. Systems based on the existing frameworks function well when dealing with small ontologies, but there are a number of limitations when dealing with larger ontologies. Some recent work (e.g. [15, 8]) has defined challenges that need to be addressed when dealing with large ontologies. According to [8] interactivity, scalability, and reasoning-based error diagnosis are required to deal with large ontologies. [15] defines the following challenges related to aligning large ontologies. Regarding scalability [15] discusses efficiency of matching techniques. This is important as many participants in the Ontology Alignment Evaluation Initiative (OAEI, a yearly event that focuses on evaluating systems that automatically generate mapping suggestions) have perfomance problems when dealing with large ontologies. Further, matching with background knowledge should be used (which could include in the [15] interpretation of background knowledge the error diagnosis of [8]). Based on OAEI experience it is also clear that there is a need for support for matcher selection, combination and tuning. There is also a need for user involvement in the matching process. The user can be involved during the mapping generation. Further, as stated by the OAEI organizers [4], automatic generation of mappings is only a first step towards a final alignment and a validation by a domain expert is needed.

In this paper we address these challenges. Our first contribution is an ontology alignment framework that enables scalability, user involvement, use of background knowledge and matcher selection, combination and tuning (Section 2). This is achieved by introducing different kinds of interruptable sessions (computation, validation and recommendation). It is the first framework that allows partial computations for generating mapping suggestions. Currently, to our knowledge, no system allows to start validating mapping suggestions before every suggestion is computed. It also is the first framework that allows a domain expert to validate a sub-set of the mapping suggestions, and continue later on. Further, it supports the use of validation results in the (re)computation of mapping suggestions and the recommendation of alignment strategies to use.

Our second contribution is the first implemented system that integrates solutions for these challenges in one system (Section 3). It is based on our session-based framework. It deals with efficiency of matching techniques by, in addition to the sessions, avoiding exhaustive pair-wise comparisons between the terms in the different ontologies. It provides solutions to matching with background knowledge by using previous decisions on mapping suggestions as well as using thesauri and domain-specific corpora. Matcher selection, combination and tuning is achieved by using an approach for recommending matchers, combinations and filters. Further, user involvement is supported in the validation phase through user interfaces that have taken into account earlier experiments with ontology engineering systems user interfaces. Also, user decisions are taken into account in the matching and recommendation steps.

Our third contribution are experiments (Section 4) that show the advantages of the session-based approach. They show alignment quality improvements based on the new functionality and show how such a system can be used for evaluating strategies that could not (easily) be evaluated before.

2 Framework

Our framework is presented in Figure 1. The input are the ontologies that need to be aligned. The output is an alignment between the ontologies which consists of a set of mappings that are accepted after validation. When starting an alignment process the user starts a computation session. When a user returns to an alignment process, she can choose to start or continue a computation session or a validation session.

During the *computation sessions* mapping suggestions are computed. The computation may involve preprocessing of the ontologies, matching, and combination and filtering of matching results. Auxiliary resources such as domain knowledge and dictionaries may be used. A reasoner may be used to check consistency of the proposed mapping suggestions in connection with the ontologies as well as among each other. Users may be involved in the choice of algorithms. This is similar to what most ontology alignment systems do. However, in this case the algorithms may also take into account the results of previous validation and recommendation sessions. Further, we allow that computation sessions can be stopped and partial results can be delivered. It is therefore possible for a domain expert to start validation of results before all suggestions are computed. The output of a computation session is a set of mapping suggestions.

During the *validation sessions* the user validates the mapping suggestions generated by the computation sessions. A reasoner may be used to check consistency of the validations. The output of a validation session is a set of mapping decisions (accepted and rejected mapping suggestions). The accepted mapping suggestions form a partial alignment (PA) and are part of the final alignment. The mapping decisions (regarding acceptance as well as rejection of mapping suggestions) can be used in future computation sessions as well as in recommendation sessions. Validation sessions can be stopped and resumed at any time. It is therefore not neccesary for a domain expert to validate all mapping suggestions in one session. The user may also decide not to resume the validation but start a new computation session, possibly based on the results of a recommendation session.

The input for the *recommendation sessions* consists of a database of algorithms for the preprocessing, matching, combination and filtering in the computation sessions. During the recommendation sessions the system computes recommendations for which (combination) of those algorithms may perform best for aligning the given ontologies. When validation results are available these may be used to evaluate the different algorithms, otherwise an oracle may be used. The output of this session is a recommendation for the settings of a future computation session. These sessions are normally run when a user is not validating and results are given when the user logs in into the system again.

Most existing systems can be seen as an instantiation of the framework with one or more computation sessions. Some systems also include one validation session.

3 Implemented System

We have implemented a prototype based on the framework described above. The system includes components from the SAMBO system and newly developed components.

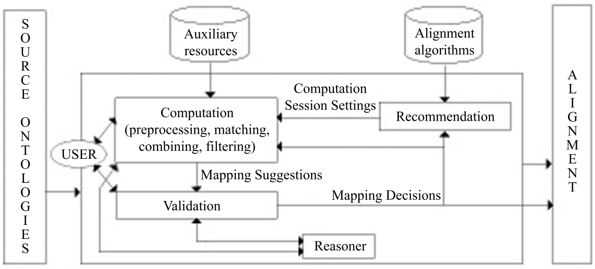
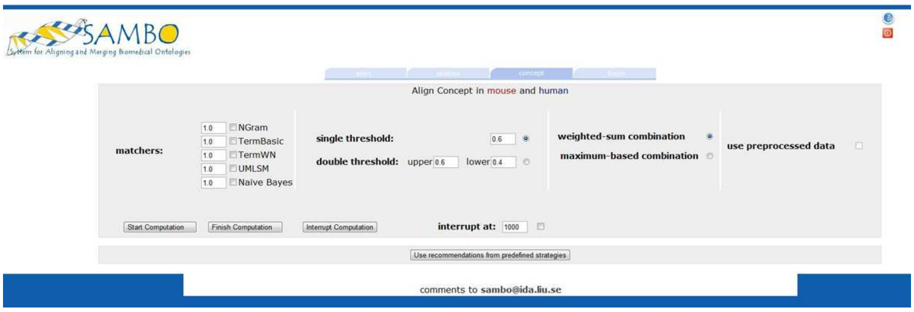


Fig. 1. Framework.


SAMBO
System for Aligning and Merging Biomedical Ontologies
start
relation
concept
finish
Start Computation
Finish Computation
Interrupt Computation
Use recommendations from predefined strategies

Align Concept in mouse and human

matchers:
1.0 □ NGram
1.0 □ TermBasic
1.0 □ TermWN
1.0 □ UMLSM
1.0 □ Naive Bayes

single threshold: 0.6 ●

double threshold: upper 0.6 lower 0.4 ○

weighted-sum combination ●

maximum-based combination ○

use preprocessed data □

interrupt at: 1000 □

comments to sambo@ida.liu.se

Fig. 2. Screenshot: start computation session.

Session Framework. When starting an alignment process for the first time, the user starts a computation session. However, if the user has previously stored sessions, then a screen is shown where the user can start a new session or resume a previous session.

The information about sessions is stored in the session management database. This includes information about the user, the ontologies, the list of already validated mappings suggestions, the list of not yet validated mappings suggestions, and last access date. In the current implementation only validation sessions can be saved. When a computation session is interrupted, a new validation session is created and this can be stored.

Computation. Figure 2 shows a screen shot of the system at the start of a computation session. It allows for the setting of the session parameters. The computation of mapping suggestions uses the following steps. During the *settings selection* the user selects which algorithms to use for the preprocessing, matching, combining and filtering steps. An experienced user may choose her own settings. Otherwise, the suggestion of a recommendation session (by clicking the ’Use recommendations from predefined strategies’ button) or a default setting may be used. This information is stored in the session information database.

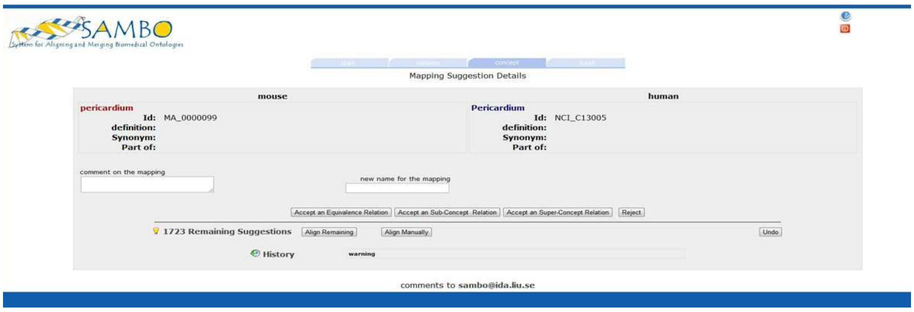
When a PA is available, the *preprocessing* step partitions the ontologies into corresponding mappable parts that make sense with respect to the structure of the ontologies (details in [11]). Therefore, the matchers will not compute similarity values between all pairs of concepts, but only between concepts in mappable parts, thereby considerably reducing the search space. The user may choose to use this preprocessing step by checking the 'use preprocessed data' check box (Figure 2).

Matchers compute similarity values between terms in different ontologies. Whenever a similarity value for a term pair using a matcher is computed, it is stored in the similarity values database. This can be done during the computation sessions, but also during the recommendation sessions. In the current implementation we have used string matching for matching relations. Regarding concepts, the matchers compute similarity values between pairs of concepts as received from the preprocessing step (all pairs or pairs of concepts in mappable parts). We use the linguistic, WordNet-based, UMLS-based and instance-based algorithms from the SAMBO system [13]. The matcher *n-gram* computes a similarity based on 3-grams. The matcher *TermBasic* uses a combination of n-gram, edit distance and an algorithm that compares the lists of words of which the terms are composed. The matcher *TermWN* extends TermBasic by using WordNet [20] for looking up is-a relations. The matcher *UMLSM* uses the domain knowledge in the Unified Medical Language System (UMLS, [17]) to obtain mappings. Finally, the instance-based matcher *NaiveBayes* makes use of research literature that is related to the concepts in the ontologies. It is based on the intuition that a similarity measure between concepts in different ontologies can be defined based on the probability that documents about one concept are also about the other concept and vice versa [18].

The user can define which matchers to use in the computation session by checking the check boxes in front of the matchers' names (Figure 2). To guarantee partial results as soon as possible the similarity values for all currently used matchers are computed for one pair of terms at a time and stored in the similarity values database. When the similarity values for each currently used matcher for a pair of terms are computed, they can be combined and filtered (see below) immediately. As ontology alignment is an iterative process, it may be the case that the similarity values for some pairs and some matchers were computed in a previous round. In this case these values are already in the similarity values database and do not need to be re-computed.

Results from different matchers can be *combined*. In our implemented system we allow the choice of a weighted-sum approach or a maximum-based approach. In the first approach each matcher is given a weight and the final similarity value between a pair of terms is the weighted sum of the similarity values divided by the sum of the weights of the used matchers. The maximum-based approach returns as final similarity value between a pair of terms, the maximum of the similarity values from different matchers. The user can choose which combination strategy to use by checking radio buttons (Figure 2), and weights can be added in front of the matchers' names.

Most systems use a threshold *filter* on the similarity values to decide which pairs of terms become mapping suggestions. In this case a pair of terms is a mapping suggestion if the similarity value is equal to or higher than a given threshold value. Another approach that we implemented is the double threshold filtering approach [1] where two thresholds are introduced. Pairs with similarity values equal to or higher than the upper threshold are retained as mapping suggestions. These pairs are also used to partition the ontologies in a similar way as in the preprocessing step. The pairs with similarity values



SAMBO
System for Aligning and Merging Biomedical Ontologies
start
relation
concept
finish

Mapping Suggestion Details

mouse

pericardium
Id: MA_0000099
definition:
Synonym:
Part of:

human

Pericardium
Id: NCI_C13005
definition:
Synonym:
Part of:

comment on the mapping

new name for the mapping

Accept an Equivalence Relation
Accept an Sub-Concept Relation
Accept an Super-Concept Relation
Reject

1723 Remaining Suggestions

Align Remaining
Align Manually
Undo
History

warning

comments to sambo@ida.liu.se

Fig. 3. Screenshot: mapping suggestion.

between the lower and upper thresholds are filtered using the partitions. Only pairs of which the elements belong to corresponding elements in the partitions are retained as suggestions. Pairs with similarity values lower than the lower threshold are rejected as mapping suggestions. When a PA is available, a variant of double threshold filtering can be used, where the PA is used for partitioning the ontologies [11]. The user can choose single or double threshold filtering and define the thresholds (Figure 2). Further, to obtain higher quality mappings, we always remove mapping suggestions that conflict with already validated correct mappings [11].

The computation session is started using the ’Start Computation’ button. The session can be interrupted using the ’Interrupt Computation’ button. The user may also specify beforehand a number of concept pairs to be processed and when this number is reached, the computation session is interrupted and validation can start. This setting is done using the ’interrupt at’ in Figure 2. The output of the computation session is a set of mapping suggestions where the computation is based on the settings of the session. Additionally, similarity values are stored in the similarity values database that can be used in future computation sessions as well as in recommendation sessions. In case the user decides to stop a computation session, partial results are available, and the session may be resumed later on. The ’Finish Computation’ button allows a user to finalize the alignment process. (A similar button is available in validation sessions.)

Validation. The validation session allows a domain expert to validate mapping suggestions. The mapping suggestions can come from a computation session (complete or partial results) or be the remaining part of the mapping suggestions of a previous validation session. For the validation we extended the user interface of SAMBO [13] which took into account lessons learned from experiments [9, 10] with ontology engineering systems’ user interfaces. As stated in [6] our user interface evaluations are one of the few existing evaluations and our system is one of the few systems based on such evaluation. Through the interface, the system presents mapping suggestions (Figure 3) with available information about the terms in the mapping suggestions. When a term appears in multiple mapping suggestions, these will be shown at the same time. The user can accept a mapping suggestion as an equivalence or is-a mapping, or reject the mapping suggestion by clicking the appropriate buttons. Further, the user can give a preferred name to equivalent terms as well as annotate the decisions. The user can also review the previous decisions ('History') as well as receive a summary of the mapping suggestions still to validate ('Remaining Suggestions'). After validation a reasoner is used to detect conflicts in the decisions and the user is notified if any such occur.

The mapping decisions are stored in the mapping decisions database. The accepted mapping suggestions constitute a PA and are partial results for the final output of the ontology alignment system. The mapping decisions (both accepted and rejected) can also be used in future computation and recommendation sessions.

Validation sessions can be stopped at any time and resumed later on (or if so desired - the user may also start a new computation session).

Recommendation. We implemented several recommendation strategies. The first approach (an extension of [19]) requires the user or an oracle to validate all pairs in small segments of the ontologies. To generate these segments we first use a string-based approach to detect concepts with similar names. The sub-graphs of the two ontologies with the matched concepts as roots are then candidate segments. Among the candidate segments a number of elements (15) of small enough size (60 concepts) are retained as segments. As a domain expert or oracle has validated all pairs in the segments, full knowledge is available for these small parts of the ontologies. The recommendation algorithm then proposes a particular setting for which matchers to use, which combination strategy and which thresholds, based on the performance of the strategies on the validated segments. The advantage of the approach is that it is based on full knowledge of the mappings of parts of the ontologies. An objection may be that good performance on parts of the ontologies may not lead to good performance on the whole ontologies. The disadvantage of the approach is that a domain expert or an oracle needs to provide full knowledge about the mappings of the segments.

The second and third approach can be used when the results of a validation are available. In the second approach the recommendation algorithm proposes a particular setting based on the performance of the alignment strategies on all the already validated mapping suggestions. In the third approach we use the segment pairs (as in the first approach) and the results of earlier validation to compute a recommendation. The advantages of these approaches are that decisions from different parts of the ontologies can be used, and that no domain expert or oracle is needed during the computation of the recommendation. However, no full knowledge may be available for any parts of the ontologies (e.g. for some pairs in the segment pairs, we may not know whether the mapping is correct or not), and validation decisions need to be available.

We note that in all approaches, when similarity values for concepts for certain matchers that are needed for computing the performance, are not yet available, these will be computed and added to the similarity values database.

To define the performance of the alignment algorithms several measures can be used. We define the measures that are used in our implementation. We assume there is a set of pairs of terms for which full knowledge is available about the correctness of the mappings between the terms in the pair. For the first approach this set is the set of pairs in the segments. In the other approaches this set is the set of pairs in the mappings decisions (accepted and rejected). For a given alignment algorithm, let then A be the number of pairs that are correct mappings and that are identified as mapping

Table 1. Performance measures.

(a) Number of correct/wrong mappings that are suggested/not suggested.

	Suggested	Not suggested
Correct	A	C
Wrong	B	D

(b) Measures.

P^c = A/(A+B), R^c = A/(A+C), F^c = $2P^cR^c/(P^c+R^c)$
P^w = D/(C+D), R^w = D/(B+D), F^w = $2P^wR^w/(P^w+R^w)$
Sim1 = (A+D)/(A+B+C+D), Sim2 = A/(A+B+C)

suggestions, B the number of pairs that are wrong mappings but were suggested, C the number of pairs that are correct mappings but that were not suggested, and D the number of pairs that are wrong mappings and that were not suggested (see Table 1(a)). In A + D cases the algorithm made a correct decision and in B + C cases the algorithm made a wrong decision. In our system we use then the following measures (see Table 1(b)). P^c, R^c and F^c are the common measures of precision, recall and their harmonic mean f-measure. These focus on correct decisions for correct mappings. P^w, R^w and F^w are counterparts that focus on correct decisions regarding wrong mappings. Sim1 is a similarity measure that computes the ratio of correct decisions over the total number of decisions. Sim2 is the Jaccard-similarity where the case of non-suggested wrong mappings is not taken into account (assumed to be a common and non-interesting case).

The results of the recommendation algorithms are stored in the recommendation database. For each of the alignment algorithms (e.g. matchers, combinations, and filters) the recommendation approach and the performance measure are stored. A user can use the recommendations when starting or continuing a computation session.

4 Experiments

In this Section we discuss experiments that show the advantages of using a session-based system regarding performance of computation of similarity values, filtering and recommendation. Further, the experiments in Sections 4.2-4.3 also show how a session-based system can be used for evaluating PA-based and recommendation algorithms.

Experiments Set-up. We use the OAEI 2011 Anatomy track for our experiments which contains the ontologies Adult Mouse Anatomy (AMA) and the anatomy part of the NCI Thesaurus (NCI-A). (Removing empty nodes in the files) AMA contains 2737 concepts and NCI-A contains 3298 concepts. This gives 9,026,626 pairs of concepts. Further, a reference alignment containing 1516 equivalence mappings is available.

Regarding the alignment strategies, we used the following. As matchers we used *n-gram*, *TermBasic*, *TermWN*, *UMLSM* and *NaiveBayes*.[1] As combination strategies we used weighted sum with possible weights 1 and 2 as well as the maximum-based approach. Further, we used the single and double threshold strategies with threshold

[1] For *NaiveBayes* we generated a corpus of PubMed abstracts. We used a maximum of 100 abstracts per concept. For AMA the total number of documents was 30,854. There were 2413 concepts for which no abstract was found. For NCI-A the total number of documents was 40,081. There were 2886 concepts for which no abstract was found.

Table 2. Top 10 strategies for F^c and Sim2.

matchers	weights	threshold	correct suggestions	wrong suggestions	F^c	Sim2
TermBasic;*UMLSM*	1;1	0.4;0.7	1223	101	0.8612	0.7563
TermWN;*UMLSM*;*NaiveBayes*;*n-gram*	1;2;2;1	0.3;0.5	1223	101	0.8612	0.7563
n-gram;*TermBasic*;*UMLSM*	1;1;2	0.5;0.8	1192	63	0.8603	0.7549
n-gram;*UMLSM*	1;1	0.5;0.8	1195	67	0.8603	0.7548
UMLSM;*NaiveBayes*;*TermWN*	2;1;2	0.4;0.6	1203	78	0.8602	0.7547
UMLSM;*NaiveBayes*;*n-gram*;*TermBasic*	2;1;1;1	0.4;0.6	1199	73	0.8601	0.7545
n-gram;*TermBasic*;*UMLSM*	1;2;2	0.5;0.8	1181	50	0.8598	0.7541
UMLSM;*NaiveBayes*;*TermBasic*	2;1;2	0.4;0.6	1194	68	0.8596	0.7537
UMLSM;*NaiveBayes*;*n-gram*;*TermBasic*	2;2;1;1	0.3;0.5	1221	104	0.8595	0.7537
UMLSM;*NaiveBayes*;*TermBasic*	2;1;1	0.5;0.6	1187	60	0.8592	0.7531

Table 3. Three alignment strategies.

strategy	matchers	weights	threshold	suggestions	F^c	Sim2
AS1	*TermBasic*;*UMLSM*	1;1	0.4;0.7	1324	0.86	0.75
AS2	*TermWN*;*n-gram*;*NaiveBayes*	2;1;1	0.5	1824	0.65	0.48
AS3	*n-gram*;*TermBasic*;*UMLSM*	1;1;2	0.3	4061	0.48	0.32

values 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8. In total this gives us 4872 alignment strategies. For each of these strategies we computed P^c, R^c, F^c, P^w, R^w, F^w, Sim1 and Sim2 based on the OAEI reference alignment. For instance, Table 2, shows the top 10 strategies with respect to Sim2. All these 10 strategies use a weighted-sum combination, double threshold and include *UMLSM* and at least one string matching-based matcher. These strategies have also a very high F^w of over 0.99. The top 10 strategies with respect to R^c all include *UMLSM* and at least one of *n-gram* or *TermWN*. All these strategies use a maximum-based combination approach, single threshold and, as expected, a low threshold (0.3). The best strategies find 1497 correct mapping suggestions. The highest P^c for these strategies is, however, less than 0.016. When sorting strategies based on P^c, 528 strategies had maximum P^c value of 1. All of these strategies include *NaiveBayes*. Six of the strategies are single matcher strategies (*NaiveBayes* with thresholds 0.6, 0.7, 0.8, 0.6;07, 0.6;0.8 and 0.7;0.8). No strategy has threshold 0.3. Among those strategies the maximum amount of correct mapping suggestions that are found is 259. All 528 strategies have R^w = 1 and $\mathrm{P}^w > 0.99$. They have high Sim1 values and low Sim2 values. With respect to the other measures, i.e. R^w, P^w, F^w and Sim1, the strategies do not show much variation. Therefore, in the remainder of this paper, we mainly discuss results with respect to F^c and Sim2. F^c is a standard measure; Sim2 has a high correlation to F^c, but has a higher degree of differentiation in our experiments.

For the experiments in Sections 4.2 and 4.3 we chose three alignment strategies (Table 3) as a basis for discussion. Strategy AS1 uses a weighted sum combination of *TermBasic* with weight 1 and *UMLSM* with weight 1, and as double thresholds 0.4;0.7.

Table 4. Matcher computation time (in mins).

	n-gram		*NaiveBayes*	
number of pairs	without previous values stored	with previous values stored	without previous values stored	with previous values stored
902,662	2.59		196.15	
1,805,324	5.08	3.98	149.95	84.05
4,513,310	12.73	10.78	418.49	265.87
6,769,965	19.19	13.83	645.71	212.35
9,026,626	25.85	17.32	790.74	207.64

This information is presented in columns 2-4 in Table 3. AS1 generates 1324 mapping suggestions (column 5). AS1 is the strategy with best F^c (0.86) and with best Sim2 (0.75) values. AS2 is an average strategy regarding F^c (0.65) and Sim2 (0.48). It uses a weighted sum combination of *TermWN* with weight 2, *n-gram* with weight 1 and *NaiveBayes* with weight 1, and as threshold 0.5. It generates 1824 mapping suggestions. AS3 performs poorly for F^c (0.48) and Sim2 (0.32), but has a high R^c value (0.89). It uses a weighted sum combination of *n-gram* with weight 1, *TermBasic* with weight 1, and *UMLSM* with weight 2, and as threshold 0.3. It generates 4061 mapping suggestions.

4.1 Computation of Similarity Values

For each of the matchers we computed the similarity values for all pairs of concepts. When a similarity value is computed it is stored in the similarity values database. Previous approaches could not take advantage of previously stored values. However, computation sessions in a session-based approach can take advantage of the fact that previous computation sessions already stored similarity values. In Table 4 we show for two of the matchers the computation times for when previous values were stored and for when no previous values were stored. We do this for the computation of 10%, 20% (of which 10% stored), 50% (of which 20% stored), 75% (of which 50% stored) and 100% (of which 75% stored) of the 9,026,626 pairs. For instance, for *n-gram* the computation and storage of 902,662 similarity values took 2.59 minutes. The computation and storage of 1,805,324 similarity values from scratch took 5.08 minutes. However, assuming 902,662 similarity values are already stored and checking the database, it will take 3.98 minutes. Using the database is advantageous for string matchers, and even more advantageous for more complex matchers for which the speed-up may be up to 25%. The session-based approach leads therefore to reduced computation times and reduced waiting times for the domain expert.

4.2 Using the Validation Decisions from Previous Sessions for Filtering

There are few approaches that can take into account already given mappings. Further, it is not common that such a set a pre-existing mappings exists. In a session-based approach, however, every validation session generates such sets, which can be used to

Table 5. Filter using validated correct mappings.

processed	AS1	AS2	AS3
500	20	107	156
1000	26	58	288
1300	4	20	20

improve the quality of the mapping suggestions and reduce unnecessary user interaction. Further, the knowledge of the domain expert is taken into account in an early stage.

Filtering using Validated Correct Mappings. Table 5 shows for the strategies AS1, AS2 and AS3 the reduction of the number of suggestions by using the filter strategy that removes mapping suggestions that are in conflict with already validated correct mappings. It shows the number of removed mapping suggestions after 500, 1000 and 1300 processed mapping suggestions. The results show that AS1 does not produce many mapping suggestions that would conflict. They also suggest that the removal should be done as soon as possible. For instance, when we would process 1000 suggestions without removal, the 156 that would be removed after 500 processed suggestions may actually have been - unnecessarily - validated by the user. Therefore, in our system we perform the removal after every validation of a correct equivalence mapping and thereby reduce unnecessary user interaction. We also remind that the strategies AS1, AS2 and AS3 produce 1365, 1824 and 4061 mapping suggestions, respectively. Therefore, having processed 1000 mapping suggestions means that 73%, 40% and 25% of the suggestions have been processed for AS1, AS2 and AS3, respectively.

Double Threshold Filtering using Validated Correct Mappings. In our second experiment, once a session is locked, we use double threshold filtering with thresholds 0.3 (lowest considered threshold) and 0.6 on the remaining unvalidated mapping suggestions of that session. Table 6 shows for the strategies AS1, AS2 and AS3 the total number of mapping suggestions (columns 2-4) and the number of correct suggestions (columns 5-7) that are removed by this operation. There are two values separated by '/'. As double threshold filtering heavily relies on the structure of the ontologies and many is-a relations are actually missing in AMA and NCI-A [12], we experimented with the original ontologies (first value) and the repaired ontologies (second value). The results show that this filtering has a positive effect on F^c. Further, in most cases more mapping suggestions, but also more correct suggestions are removed in the original ontologies than in the repaired ontologies, and the quality in terms of F^c is higher for the repaired ontologies. We also note that for the best strategy the effect is not that high.

4.3 Recommendation Strategies with and without Sessions

For the recommendation experiments we used Sim2 as recommendation measure. For some of the experiments we also needed to generate segment pairs. The system generated 94 segment pair candidates of which 15 were randomly chosen as segment pairs. The maximum number of concepts in a segment is 12 and the minimum number is 3. The total number of concept pairs for all 15 segment pairs together is 424. According to

Table 6. Double threshold filter using validated correct mappings.

processed	AS1 suggestions removed	AS2 suggestions removed	AS3 suggestions removed	AS1 correct removed	AS2 correct removed	AS3 correct removed
500	0/2	134/113	244/279	0/0	12/1	9/1
1000	1/0	52/47	532/470	1/0	1/0	22/4
1300	0/2	43/35	443/276	0/0	9/2	21/3

the reference alignment of the OAEI, 46 of those are correct mappings. The maximum number of correct mappings within a segment pair is 7 and the minimum is 1.

Session-based Recommendation using Validation Decisions Only. In this experiment we use the recommendation algorithm that computes a performance measure for the alignment strategies based on how the strategies perform on the already validated mapping suggestions. Table 7, rows 'rec1', show the recommended strategies together with their F^c value on the current validation decisions and their actual F^c value, after having processed 500/503[2], 1000, ..., 4000 suggestions for AS1 and AS3, respectively. For AS1, AS1 itself does not appear among the top 10 recommendations for all the sessions. The strategies that received the best score for 500, 1000 and 1300 processed suggestions have actual F^c values of 0.18, 0.85 and 0.23 respectively. The results are explained by the consistent group in the double threshold filtering. For AS3, the strategy that receives the best score after 1000, 2000 and 2500 processed suggestions is also the best strategy (AS1) in reality. Otherwise, AS1 is within the top 10 recommendations. In these cases AS1 is not recommended because it suggests 2, 1, 13, 6 and 48 wrong mapping suggestions for 503, 1500, 3000, 3500 and 4000 processed suggestions, respectively, which are not suggested by the recommended strategy. The reason for the better performance of the recommended strategy is due to the generated consistent group in the double threshold filtering. We note that the recommended strategy always has an actual $F^c \geq 0.85$ (with best 0.861 for AS1).

Further, we performed an experiment where a recommendation was computed after every 500 validations and every time the recommended strategy was used. We noted that usually the recommendations improved. For instance, when using the recommended strategy after 500 validations for AS1 for computing the next 500 suggestions, leads to an improved recommendation after the 500 new suggestions are validated.

Session-based Recommendation using Segment Pairs and Validation Decisions. In this experiment we use the recommendation algorithm that uses segment pairs and computes a performance measure for the alignment strategies based on how the strategies perform on the already validated parts of the segment pairs. Table 7, rows 'rec2', show the results for AS1 and AS3, respectively. For AS1, the recommended strategy after 500, 1000 and 1300 processed suggestions has actual F^c = 0.07. The reason for this result is that AS1 has very high precision so the oracle (validated suggestions) has very little information about wrong mapping suggestions. However, it has much information about correct mapping suggestions. The strategy that is recommended in the three ses-

[2] 503, because the validation decision for suggestion 500 removes other suggestions.

Table 7. Recommendations for AS1 and AS3.

	processed suggestions	rec	matchers	weights	threshold	rec F^c	actual F^c
AS1	500	rec1	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;1;2;1	0.3;0.6	0.993	0.186
		rec2	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
	1000	rec1	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	2;1;2;1	0.5;0.7	0.992	0.850
		rec2	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
	1300	rec1	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;2;1	0.3;0.7	0.972	0.235
		rec2	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
AS3	503	rec1	*n-gram* ;*TermBasic*;*UMLSM*	1;1;2	0.4;0.8	0.920	0.850
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	1000	rec1	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.950	0.861
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	1500	rec1	*TermBasic*;*UMLSM*;*TermWN*	1;2;1	0.4;0.7	0.940	0.860
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	2000	rec1	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.920	0.861
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	2500	rec1	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.920	0.861
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
	3000	rec1	*UMLSM*;*TermWN*	1;1	0.4;0.7	0.920	0.860
		rec2	*n-gram* ;*TermBasic*;*TermWN*;*UMLSM*; *NaiveBayes*	1;1;1;2;1	0.3;0.7	1	0.760
	3500	rec1	*UMLSM*;*NaiveBayes*;*n-gram* ;*TermBasic*	2;2;1;1	0.3;0.5	0.920	0.860
		rec2	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;1	0.3;0.6	1	0.820
	4000	rec1	*n-gram* ;*TermBasic*;*UMLSM*	1;1;2	0.5;0.8	0.920	0.860
		rec2	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;1	0.3;0.6	0.990	0.820

sions is one that has very high recall but that also suggests many wrong mapping which the algorithm cannot detect. For AS3, the strategies that are recommended after 503, 1000, 1500, 2000 and 2500 processed suggestions have actual F^c = 0.53, after 3000 actual F^c = 0.76, and after 3500 and 4000 actual F^c = 0.82. This result shows that as the number of processed suggestions increases, the recommended strategy becomes better. This is because the quality of the oracle increases.

Session-independent Recommendation using Segment Pairs and Oracle. In this experiment we use the recommendation algorithm that uses segment pairs and computes a performance measure for the alignment strategies based on how the strategies perform on the segment pairs. This requires an oracle that has full knowledge about the mappings in the segment pairs and for this we use the reference alignment as provided by the OAEI. As this recommendation strategy is independent from the actual validation decisions, the recommendation does not change during the alignment process. It can therefore be performed in the beginning. Based on the performance on the 15 small segments pairs (with a reference alignment of only 46 mappings), the recommendation algorithm gives Sim2 = 0.87 and F^c = 0.93 for AS1, Sim2 = 0.52 and F^c = 0.68 for AS2, and Sim2 = 0.47 and F^c = 0.64 for AS3.

However, there are also 145 strategies that have a higher Sim2 value than AS1. The top 8 recommended strategies all use double threshold filtering and have Sim2 = 0.98 and F^c = 0.99 for the segment pairs, and an actual F^c between 0.8 and 0.84. They suggest 45 correct mappings and 0 wrong mappings, whereas AS1 suggests 42 correct mappings and 2 wrong mappings. We also note that that there are 81 strategies which have Sim2 $>$0.9 and F^c $>$0.95 on the segment pairs.

5 Related Work

To our knowledge there is no other framework or system that deals with all the challenges for aligning large ontologies that our approach deals with. Many systems generate mapping suggestions and can be seen as covering a computation session. This is also what is evaluated at the OAEI. There are some systems that allow validation of mappings such as SAMBO [13], COGZ [7] for PROMPT, and COMA++ [2]. None of these systems allow, however, interruptable sessions. LogMap2 [8] allows user interaction although it does not have graphical user interfaces yet. Interrupting user interaction in this case means using heuristics to deal with remaining mapping suggestions. Regarding the computation session components of our system, many matchers have been proposed (e.g. many papers on http://ontologymatching.org/). There are some approaches that reduce the search space by segmenting or partitioning the ontologies [2, 16]. The main difference with our approach is that we use validation decisions to partition the ontologies. Our combination strategies are standard strategies. Most systems use single threshold filtering, while we also allow double threshold filtering. There are very few recommendation approaches. The approach in [3] proposes a machine learning approach to optimize alignment strategies and is complementary to our approach. Further, there are approaches that try to minimize user interaction (e.g. [14]).

6 Conclusion

In this paper we presented to our knowledge the first framework and implemented system that allows a user to interrupt and resume the different stages of the ontology alignment task. Our work addressed several of the challenges in ontology alignment [15].

Further, we showed the usefulness of the system and its components through experiments with many alignment strategies on the OAEI 2011 Anatomy track ontologies. We showed that we obtain better quality suggestions using the session-based approach. For instance, one of the lessons learned from the experiments is that filtering after the locking of sessions is useful and the worse the initial strategy, the more useful this is. Better quality suggestions are also achieved through the use of validated mappings in the preprocessing phase. In all these cases domain expert knowledge is taken into account through the validated mappings. We also showed that the use of the session-based approach reduces unnecessary user interaction. Further, the recommendation is important, especially when the initial strategy is not good. It is also clear that the approaches using validation decisions, become better the more suggestions are validated. For the approaches using segment pairs, the choice of the segment pairs influences the recommendation results (which is different from the conclusions of experiments in [19]).

We note that the session-based framework enabled experimentation and evaluation of new alignment approaches (both in computation and recommendation) that are based on validation decisions. These evaluations were not possible or cumbersome before.

In future work we will continue to develop and evaluate computation strategies and recommendation strategies. Especially interesting are strategies that reuse validation results to e.g. reduce the search space or guide the computation. Further, we will investigate new strategies for recommendations using validation decisions.

Acknowledgements. We thank Qiang Liu, Muzammil Zareen Khan and Shahab Qadeer for their implementation work on earlier versions of the system.

References

1. B Chen, P Lambrix, and H Tan. Structure-based filtering for ontology alignment. In *IEEE WETICE Workshop on Semantic Technologies in Collaborative Applications*, pages 364–369, 2006.
2. H-H Do and E Rahm. Matching large schemas: approaches and evaluation. *Information Systems*, 32:857–885, 2007.
3. M Ehrig, S Staab, and Y Sure. Bootstrapping ontology alignment methods with APFEL. In *4th International Semantic Web Conference, LNCS 3729*, pages 186–200, 2005.
4. J Euzenat, C Meilicke, H Stuckenschmidt, P Shvaiko, and C Trojahn. Ontology alignment evaluation initiative: Six years of experience. *Journal om Data Semantics*, XV:158–192, 2011.
5. J Euzenat and P Schvaiko. *Ontology Matching*. Springer, 2007.
6. S Falconer and N Noy. Interactive techniques to support ontology matching. In *Schema Matching and Mapping*, pages 29–52. 2011.
7. S Falconer and M-A Storey. A cognitive support framework for ontology mapping. In *6th International Semantic Web Conference and 2nd Asian Semantic Web Conference, LNCS 4825*, pages 114–127, 2007.
8. E Jimenez-Ruiz, B Cuenca-Grau, Y Zhou, and I Horrocks. Large-scale interactive ontology matching: Algorithms and implementation. In *20th European Conference on Artificial Intelligence*, pages 444–449, 2012.
9. P Lambrix and A Edberg. Evaluation of ontology merging tools in bioinformatics. In *Pacific Symposium on Biocomputing*, pages 589–600, 2003.
10. P Lambrix, M Habbouche, and M Perez. Evaluation of ontology development tools for bioinformatics. *Bioinformatics*, 19(12):1564–1571, 2003.
11. P Lambrix and Q Liu. Using partial reference alignments to align ontologies. In *6th European Semantic Web Conference, LNCS 5554*, pages 188–202, 2009.
12. P Lambrix, Q Liu, and H Tan. Repairing the missing is-a structure of ontologies. In *4th Asian Semantic Web Conference, LNCS 5926*, pages 76–90, 2009.
13. P Lambrix and H Tan. SAMBO - a system for aligning and merging biomedical ontologies. *Journal of Web Semantics*, 4(3):196–206, 2006.
14. P Rodler, K Shchekotykhin, Ph Fleiss, and G Friedrich. Rio: Minimizing user interaction in debugging of aligned ontologies. In *7th International Workshop on Ontology Matching*, pages 49–60, 2012.
15. P Schvaiko and J Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, 2013.
16. M Hanif Seddiqui and M Aono. An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Journal of Web Semantics*, 7(4):344–356, 2008.

17. Unified Medical Language System. http://www.nlm.nih.gov/research/umls/about_umls.html.
18. H Tan, V Jakoniene, P Lambrix, J Aberg, and N Shahmehri. Alignment of biomedical ontologies using life science literature. In *International Workshop on Knowledge Discovery in Life Science Literature, LNBI 3886*, pages 1–17, 2006.
19. H Tan and P Lambrix. A method for recommending ontology alignment strategies. In *6th International Semantic Web Conference and 2nd Asian Semantic Web Conference, LNCS 4825*, pages 494–507, 2007.
20. WordNet. http://wordnet.princeton.edu/.