A session-based ontology alignment approach enabling user involvement

Patrick Lambrix and Rajaram Kaliyaperumal

A Session-based Ontology Alignment Approach enabling User Involvement

Patrick Lambrix [a,*] and Rajaram Kaliyaperumal [b]
[a] *Department of Computer and Information Science*
and Swedish e-Science Research Centre
Linköping University
SE-581 83 Linköping, Sweden
[b] *Department of Computer and Information Science*
Linköping University
SE-581 83 Linköping, Sweden

Abstract.
One of the current challenges in ontology alignment is the user involvement in the alignment process. To obtain high-quality alignments user involvement is needed for validation of matching results as well as in the mapping generation process. Further, there is a need for supporting the user in tasks such as matcher selection, combination and tuning.

In this paper we introduce a conceptual ontology alignment framework that enables user involvement in a natural way. This is achieved by introducing different kinds of interruptible sessions. The framework allows partial computations for generating mapping suggestions, partial validations of mapping suggestions, recommendations for alignment strategies as well as the use of validation decisions in the (re-)computation of mapping suggestions and the recommendations. Further, we show the feasibility of the approach by implementing a session-based version of an existing system. We also show through experiments the advantages of our approach for ontology alignment as well as for evaluation of ontology alignment strategies.

Keywords: Ontologies, Ontology engineering, Ontology alignment, User involvement

1. Introduction [1]

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 concepts (and relations) 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 concepts and relations in different ontologies (e.g., [11]). The research field of ontology alignment is very active with its own yearly workshop as well as a yearly event, the Ontology Alignment Evaluation Initiative (OAEI, e.g., [10]), that focuses on evaluating systems that automatically generate mapping suggestions. Many systems have been built and overviews can be found in e.g., [22,39,47,11,30,48] and at the ontology matching web site http://www.ontologymatching.org.

One of the current challenges in ontology alignment is the user involvement in the alignment process

[*] Corresponding author. E-mail: patrick.lambrix@liu.se.
[1] This paper is an extended version of [26] with additional algorithms, a user interface evaluation and additional experiments.

[48,19]. Based on OAEI experience it is 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. First, the user could be involved during the mapping generation. Introducing domain expert knowledge already in the generation phase could significantly improve the matching results and is essential for use cases requiring very accurate mappings [21]. Further, as stated by the OAEI organizers [10], automatic generation of mappings is only a first step towards a final alignment and a validation by a domain expert is needed to obtain high-quality alignments.

In this paper we introduce a conceptual *ontology alignment framework* that enables user involvement in a natural way (Section 3). Existing frameworks for ontology alignment systems (e.g., [9,31]) describe different components and steps in the ontology alignment process such as preprocessing, matching, filtering and combining match results. The user involvement in these frameworks usually relates to the validation of the mapping suggestions generated by the ontology alignment system. In this paper we introduce a, for the ontology alignment community, novel framework based on interruptible sessions: computation, validation and recommendation sessions. It is the first framework that allows partial computations for generating mapping suggestions, thereby reducing the waiting time for domain experts. 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, thereby allowing the interleaving of computation and validation. Further, it supports the use of validation results in the (re)computation of mapping suggestions and the recommendation of alignment strategies to use, thereby introducing the domain expert's knowledge in the mapping generation and recommendation processes. The framework also introduces recommendation sessions that generate recommendations for matcher selection, combination and tuning.

Further, we show the feasibility of the session-based framework by implementing a session-based version of an existing *ontology alignment system* (Section 4). We note that we are not aiming to build the 'best' possible ontology alignment system, but we want to reuse as many components as possible from an existing system (in this case SAMBO [31]), thereby showing the feasibility of extending existing systems to fit the session-based approach.[2] However, as a side effect of implementing our session-based system, we do introduce some new algorithms.

We also provide several *experiments* (Section 5). The experiments relate to some of the main features of the session-based approach (interruptable sessions, use of validation decisions from previous validation sessions, and recommendation sessions). All experiments show the advantages of *using a session-based system* as compared to a 'traditional' approach for ontology alignment. We point to alignment quality improvements based on the new functionality that the session-based approach enables in terms of performance of computation of similarity values, filtering and recommendation. Some of the experiments additionally show how a session-based system can be used for *evaluating strategies* (partial alignment-based algorithms and recommendation algorithms) that could not or not easily be evaluated before.

In Section 6 we discuss related work and Section 7 concludes the paper. First, however, we introduce some background in Section 2.

2. Background

In general, from a knowledge representation point of view, ontologies may contain concepts, relations, axioms and instances. Concepts and relations are often organized in hierarchies using the is-a (or subsumption) relation, denoted by $\sqsubseteq$. The task of ontology alignment is to create an alignment between ontologies. An alignment is a set of mappings (also called correspondences) between entities from the different ontologies. The most common kinds of mappings are equivalence mappings ($\equiv$) as well as mappings using is-a and its inverse ($\sqsubseteq$, $\sqsupseteq$). For instance, for concepts A from the first ontology and A' from the second ontology $<$A,A',$\equiv$$>$ represents the fact that A and A' are equivalent.

2.1. Ontology alignment framework

A large number of ontology alignment systems have been developed. Many of these are based on the com-

[2] For instance, we use the matchers from SAMBO as of 2006. The best SAMBO strategy regarding f-measure was still better than the best system at OAEI Anatomy 2009 and earlier, but in 2010 AgreementMaker implemented a strategy with f-measure 0.877. In 2014 and 2015, AML reached an f-measure of 0.944.

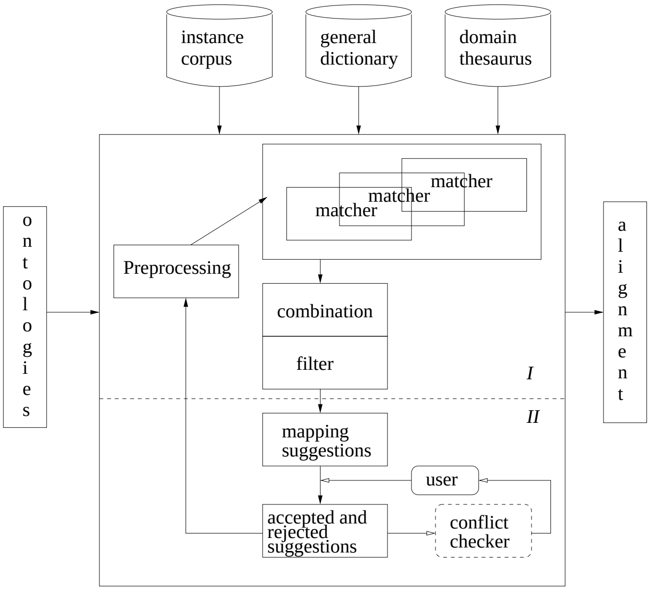


Fig. 1. An existing framework (extension of the framework in [31]).

putation of similarity values between entities in different ontologies and can be described as instantiations of the general framework in Figure 1. The framework consists of two parts. The first part (*I* in Figure 1) computes mapping suggestions. The second part (*II*) interacts with the user to decide on the final alignment.

An alignment algorithm receives as input two source ontologies. Part I typically contains different components. A preprocessing component can be used to modify the original ontologies, e.g., to extract specific features of the concepts in the ontologies, or to partition the ontologies into mappable parts thereby reducing the search space for finding mapping suggestions. The algorithm can include several matchers that calculate similarities between the entities from the different source ontologies or mappable parts of the ontologies. They often implement strategies based on linguistic matching, structure-based strategies, constraint-based approaches, instance-based strategies, strategies that use auxiliary information or a combination of these. Each matcher utilizes knowledge from one or multiple sources. Mapping suggestions are then determined by combining and filtering the results generated by one or more matchers. Common combination strategies are the weighted-sum and the maximum-based strategies. The most common filtering strategy is the (single) threshold filtering. By using different preprocessing, matching, combining and filtering techniques, we obtain different alignment strategies. The result of part I is a set of mapping suggestions.[3]

In part II the mapping suggestions are then presented to the user, a domain expert, who accepts or rejects them. The accepted mapping suggestions are part of the final alignment. Any sub-set of the final alignment is a partial alignment (PA). The acceptance and rejection of suggestions may also influence further suggestions. Further, a conflict checker could be used to avoid conflicts introduced by the mapping suggestions.[4]

There can be several iterations of parts I and II. The output of the alignment algorithm is a set of mappings between entities from the source ontologies. All systems implement part I while some also implement part II and allow iterations.

In Section 3 we propose a framework that includes the existing framework in some of its components.

[3] Traditionally, in the OAEI it is this result (and thus part I) that is evaluated. In 2013, for the first time there was a track for evaluating interaction and thus also some issues related to part II.

[4] During the recent years some systems allow not only for conflict checking but also for repairing of mappings or mapping suggestions, e.g., [35,21,37,41,18].

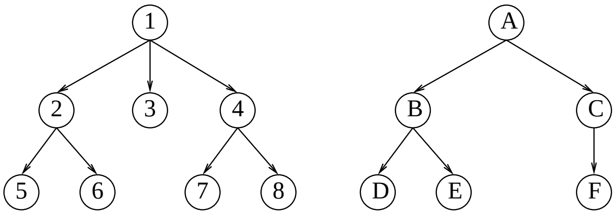


Fig. 2. Ontologies.

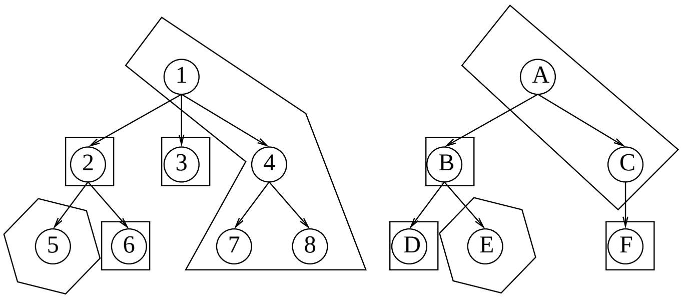


Fig. 3. Partitions.

2.2. Preliminaries

In the implemented system (Section 4) we use algorithms that require some new notions. We define these in this section.

2.2.1. Partitioning using a consistent group

Given a set of equivalence mappings M between two ontologies, a *consistent group* is a sub-set of M such that each concept occurs at most once as first argument in a mapping in M, at most once as second argument in a mapping in M and for each pair of mappings <A,A',≡> and <B,B',≡> in M where A and B are concepts in the first ontology and A' and B' are concepts in the second ontology, we require that A $\sqsubseteq$ B iff A' $\sqsubseteq$ B'. Different choices can be made for M. In the preprocessing technique in Section 4.2.2 and in one of the filtering techniques in Section 4.2.5 M is a PA. In another filtering technique in Section 4.2.5 M is a set of mapping suggestions that are not yet validated.

As an example, consider the two ontologies in Figure 2 where the nodes represent concepts and the edges inverses of is-a relations (e.g., the concept represented by node 2 is a sub-concept of the concept represented by node 1). Assume we have a PA that contains <2,B,≡>, <3,F,≡>, <6,D,≡>, and <5,C,≡>. Then the set of mappings {<2,B,≡>, <3,F,≡>, <6,D,≡>, <5,C,≡>} is not a consistent group as the concept represented by 5 is a sub-concept of the concept represented by 2, but the concept represented by C is not a sub-concept of the concept represented by B. However, {<2,B,≡>, <3,F,≡>, <6,D,≡>} is a consistent group.

Given a set of equivalence mappings M between two ontologies, finding *a* consistent group is easy. Finding a maximal consistent group, i.e., a consistent group for which no proper superset is a consistent group, is an expensive operation. Therefore, in our implementations we use a genetic algorithm that guarantees to find a consistent group, but although we usually find large consistent groups, they may not always be maximal [3].

A consistent group respects the is-a hierarchy in the two ontologies and can be used to partition the two ontologies such that each element (which is a set of concepts) in the partition of the first ontology has a corresponding element (which is a set of concepts) in the partition of the second ontology and only mappings between concepts in corresponding elements respect the structure of the ontologies. This can be done as follows. A mapping <A,A',≡> divides the two ontologies into three parts. The first ontology is divided into (i) the descendants of A, (ii) A and (iii) the rest. The second ontology is divided into (i) the descendants of A', (ii) A' and (iii) the rest. We use each mapping in the consistent group in this way. By taking the intersections of all these parts, we obtain different pieces in the first ontology that have a corresponding piece in the second ontology. Such corresponding pieces we call *mappable parts*.

As an example, consider the ontologies in Figure 2 and the consistent group {<2,B,≡>, <3,F,≡>, <6,D,≡>}. In this case the first ontology has the following pieces: descendants of node 6 (empty), node 6, descendants of node 2 that are not node 6 or descendants of node 6 (node 5), node 2, descendants of node 3 (empty), node 3, the rest (nodes 1, 4, 7, 8). The second ontology is divided into: descendants of node D (empty), node D, descendants of node B that are not node D or descendants of node D (node E), node B, descendants of node F (empty), node F, the rest (nodes A, C). The corresponding mappable parts between the ontologies are: ({5}, {E}) and ({1,4,7,8}, {A,C}) (Figure 3).

2.2.2. Segments

The intuition behind a segment of an ontology is that it represents a piece of the ontology. Formally, we define a *segment* of an ontology as a set of concepts in the ontology (usually a proper sub-set of the set of concepts in the ontology). In several of the implemented recommendation algorithms (Section 4.4) we require full knowledge about the mappings between segments from different ontologies (segment pairs). In [52] different strategies for generating segment pairs are described (e.g., based on sub-graphs of the ontologies or using clustering algorithms).

3. Alignment Framework

3.1. Framework

Our new framework is presented in Figure 4. 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. The framework defines three kinds of sessions: computation, validation and recommendation sessions. 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 (as in part I of the old framework). 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 (as in part II in the old framework). 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 interrupted and partial results can be delivered. It is therefore possible for a domain expert to start validation of results before all mapping suggestions are computed. The output of a computation session is a set of mapping suggestions.

During the *validation sessions* the domain expert 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 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 interrupted 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[5] may be used. The out-

[5] Often the oracle would be a domain expert. In the framework we do not assume any properties of the oracle, but it is clear that the quality of the oracle/domain expert has an influence on the quality of the recommendation in a similar way as it has on the validation. In [32] different kinds of oracles representing different levels of user knowledge were discussed. Experiments in [21] for the LogMap system suggest that as long as the error rate of an oracle is less than 30%, validating the mapping suggestions is beneficial. Further, the OAEI Interactive track in 2015 introduced validations with error rates.

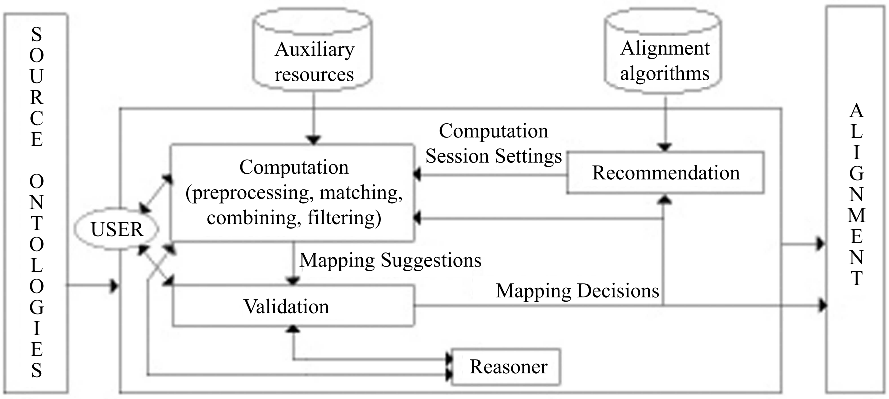


Fig. 4. Framework.

put 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.

3.2. *Coverage*

The framework covers different kinds of existing systems. For most tracks in the OAEI, participating systems usually compute an alignment using preprocessing, matching, combining and filtering algorithms. This is essentially a process in our framework where only one non-interrupted computation session is used and no validation nor recommendation sessions.

A slightly more complex variant is where the results of the computation session are validated by a domain expert. This conforms to one non-interrupted computation session and one non-interrupted validation session. This case is covered by systems with a user interface (e.g., SAMBO [31], AlViz [33], COGZ [13], COMA++ [6], AgreementMaker [5], LogMap [21], AML [42]).

Some systems use an iterative approach where validation decisions can be used in a new iteration of the computation. The validation decisions could be used to prune the search space, in matching algorithms (e.g., structure-based), or in filtering approaches. This case requires iterations of a computation session followed by a validation session.

Some systems allow saving and loading alignments (e.g., [33,6,13,5,18]). Although not session-based, it can be seen as if they mimic sessions by storing the results of a run of the system (which can be seen as computation and validation sessions) and in a new run of the system these results can be used (mimicing a new computation session).

LogMap [21] allows to interrupt validation sessions and the system then automatically deals with the remaining mapping suggestions.

In addition to covering the process of many current systems, the proposed framework also supports new and additional workflows for ontology alignment systems. A typical[6] workflow for an alignment process would be to start a computation session, possibly with a default setting for the preprocessing, matching, combination and filtering steps. After a number of mapping suggestions are computed, the computation session is interrupted and a domain expert starts validating the mapping suggestions. In case there are too many suggestions to handle in one time or when the domain expert would like to see the validation decisions up to now taken into account in the computation of the mapping suggestions, the validation session is interrupted. Then, a recommendation session computes a recommended setting for the algorithms used in the computation session and the user can decide to run a new computation session with the recommended setting or the old setting. The new computation session can also take the domain expert's validation decisions into account. This iterative process continues until no more new mapping suggestions are generated.

[6] The framework actually allows for a more flexible interleaving of the different kinds of sessions than shown in this typical workflow.

In Section 4.6 we give an example run of a system implemented based on the framework.

4. Implemented System

We have implemented a prototype based on the framework described above. We have used and extended some components from the SAMBO system [31], previously developed in our group, and developed and implemented several new components.[7]

4.1. Support for sessions

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 as in Figure 5 is shown and 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 mapping suggestions, the list of not yet validated mapping 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. When a user ends or interrupts a session, the user can ask the system to, using the obtained validation decisions, filter the non-validated mapping suggestions, preprocess the data for a future session or compute a recommendation for the settings of a new computation sesssion.

4.2. Computation sessions

4.2.1. Settings selection

Figure 6 shows a screenshot of the system at the start of a computation session. It allows for the setting of the session parameters. During the *settings selection* the user selects algorithms for the matching, combining and filtering steps as well as whether preprocessed data should be used. 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. It is also possible to inspect a list of predefined strategies as well as a list of the top recommended strategies with their recommendation scores and select a strategy from these lists. The settings selection is stored in the session information database. The computation session is started using the 'Start Computation' button.

4.2.2. Preprocessing

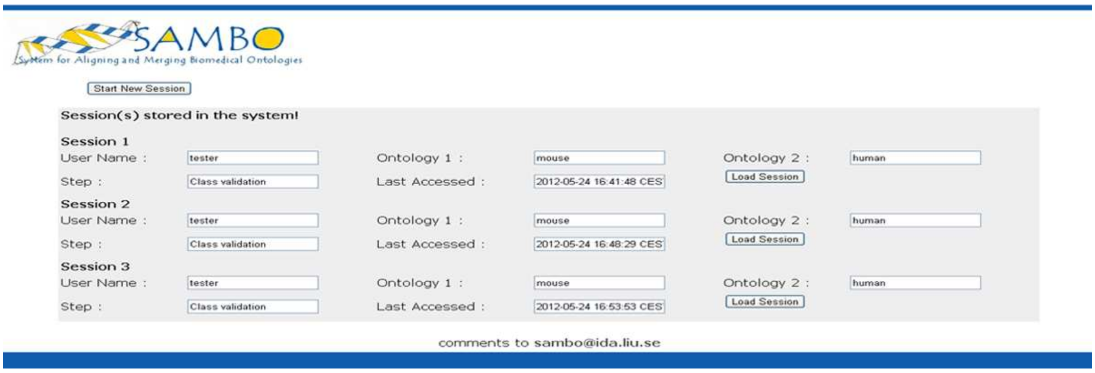
When a PA is available (e.g., after an (interrupted) validation session - in this case this step can be initiated after the end or interruption of a previous validation session), the *preprocessing* step partitions the ontologies into corresponding mappable parts according to the method we developed in [27]. This method computes corresponding mappable parts that make sense with respect to the is-a structure of the ontologies as described in Section 2.2.1. As only mappings between concepts in mappable parts respect the structure of the ontologies, 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 method has good performance when the is-a structure of the ontologies is correct. The user may choose to use this preprocessing step by checking the 'use preprocessed data' check box (Figure 6).

4.2.3. Matchers

Matchers compute similarity values between entities in different ontologies. Whenever a similarity value for an entity 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 [31]. The matcher *n-gram* computes a similarity based on 3-grams. An n-gram is a set of n consecutive characters extracted from a string. Similar strings have a high proportion of n-grams in common. The matcher *TermBasic* uses a combination of n-gram, edit distance and an algorithm that compares the lists of words of which the names of the concepts and relations are composed.[8] A Porter stemming algorithm is employed to

[7] In the text we explicitly mention which components are taken from or further developed from previous work. When nothing is mentioned, it means we have developed new algorithms.

[8] This is similar to a combination of n-gram, edit distance and Jaccard. According to [2] this should give good results for the f-measure for standard ontologies. Also according to [2], for biomedical on-



SAMBO
SyStem for Aligning and Merging Biomedical Ontologies

Start New Session

Session(s) stored in the system!

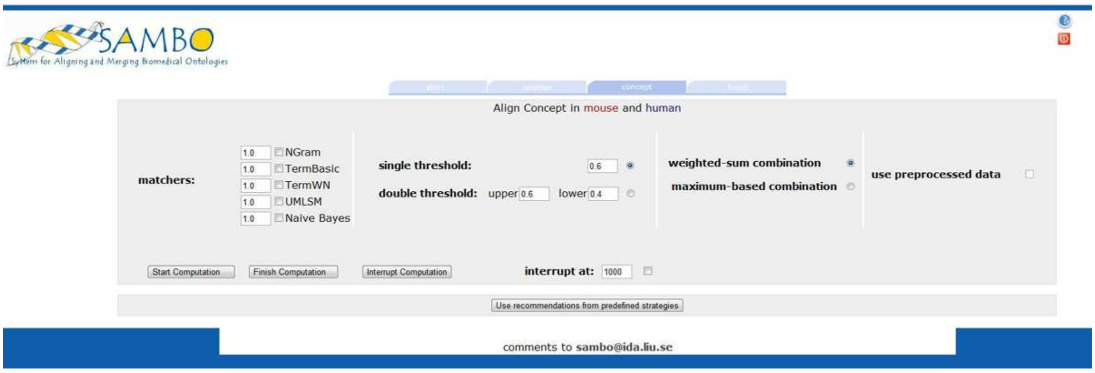
Session 1
User Name : tester
Step : Class validation
Ontology 1 : mouse
Last Accessed : 2012-05-24 16:41:48 CES
Ontology 2 : human
Load Session

Session 2
User Name : tester
Step : Class validation
Ontology 1 : mouse
Last Accessed : 2012-05-24 16:48:29 CES
Ontology 2 : human
Load Session

Session 3
User Name : tester
Step : Class validation
Ontology 1 : mouse
Last Accessed : 2012-05-24 16:53:53 CES
Ontology 2 : human
Load Session

comments to sambo@ida.liu.se

Fig. 5. Screenshot: start session.



SAMBO
SyStem for Aligning and Merging Biomedical Ontologies

start | relation | concept | finish

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 □

Start Computation | Finish Computation | Interrupt Computation | interrupt at: 1000 □

Use recommendations from predefined strategies

comments to sambo@ida.liu.se

Fig. 6. Screenshot: start computation session.

each word. The matcher *TermWN* extends TermBasic by using WordNet [55] for looking up is-a relations. The matcher *UMLSM* uses the domain knowledge in the Unified Medical Language System (UMLS, [50]) to obtain mapping suggestions. 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 can be defined based on the probability that documents about one concept are also about the other concept and vice versa [51]. For this matcher for each ontology that we want to align we generate a corpus of documents. Then for each ontology a document classifier is generated using its corpus. This classifier returns for a given document the concept that is most closely related. Documents of one ontology are then classified by the document classifier of the other ontology and vice versa and a similarity measure between concepts in the different ontologies is computed based on the number of documents related to one concept being classified to the second concept and vice versa.

The user can define which matchers to use in the computation session by checking the check boxes in front of the matchers' names (Figure 6). To guarantee partial results as soon as possible the similarity values for all currently used matchers are computed for one pair of entities at a time and stored in the similarity values database. When the similarity values for each currently used matcher for a pair of entities 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.

4.2.4. *Combining*

Results from different matchers can be *combined*. In our system we allow the choice of the two most

tologies edit distance gives good precision while Jaccard gives good recall and f-measure.

common approaches: a weighted-sum approach and a maximum-based approach. In the first approach each matcher is given a weight and the final similarity value between a pair of entities 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 entities, the maximum of the similarity values from different matchers. The user can choose which combination strategy to use by checking radio buttons, and weights can be added in front of the matchers' names (Figure 6).

4.2.5. Filtering

Most systems use a threshold *filter* on the similarity values to decide which pairs of entities become mapping suggestions. In this case a pair of entities 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 that we developed in [3]. In this approach 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 as described in Section 2.2.1, similar to the partitioning in the preprocessing step. The pairs with similarity values 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 [27]. The user can choose single or double threshold filtering and define the thresholds (Figure 6). Further, to obtain higher quality mappings, we always remove mapping suggestions that conflict with already validated correct mappings [27].

4.2.6. Ending and interrupting

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' field (Figure 6). 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 interrupt 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.)

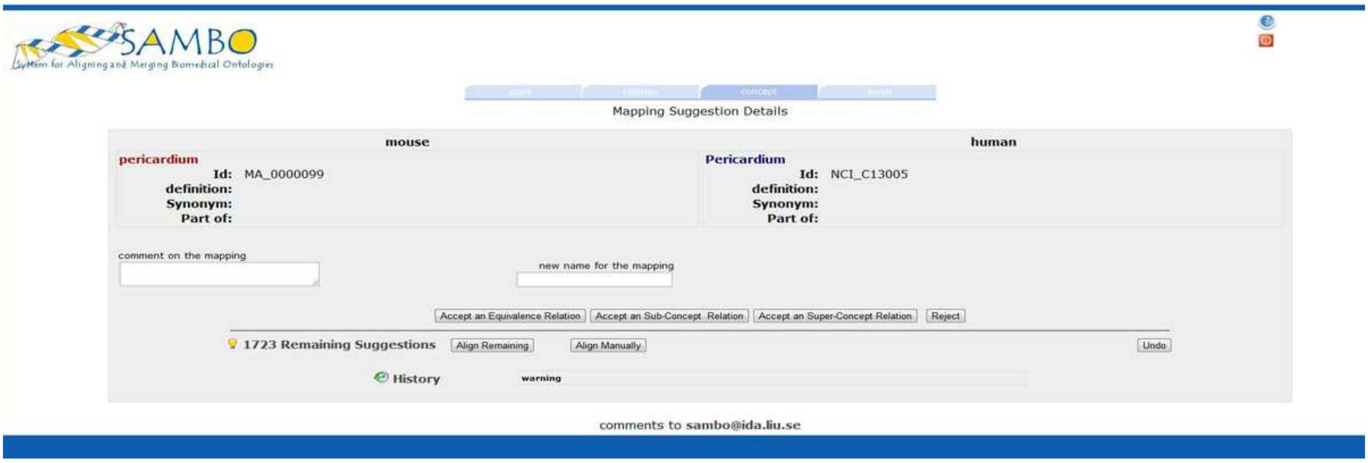
4.3. Validation sessions

The validation sessions allow 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 [31], which took into account lessons learned from experiments [23,24] with ontology engineering systems' user interfaces. As stated in [12] 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 7) with available information about the entities in the mapping suggestions. When an entity appears in multiple mapping suggestions, these will be shown at the same time. The user can accept a mapping suggestion as an $\equiv$, $\sqsubseteq$ or $\sqsupseteq$ mapping, or reject the mapping suggestion by clicking the appropriate buttons. Further, the user can give a preferred name to equivalent entities 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).

4.4. Recommendation sessions

We implemented several recommendation strategies. The first approach (an extension of our work in [52]) requires the user or an oracle to validate all pairs in small segment pairs of the different ontologies (Sec-


SAMBO
System for Aligning and Merging Biomedical Ontologies
start
relation
concept
finish
Accept an Equivalence Relation
Accept an Sub-Concept Relation
Accept an Super-Concept Relation
Reject
Align Remaining
Align Manually
Undo
History

Mapping Suggestion Details

mouse	human
pericardium	Pericardium
Id: MA_0000099	Id: NCI_C13005
definition:	definition:
Synonym:	Synonym:
Part of:	Part of:

comment on the mapping

new name for the mapping

1723 Remaining Suggestions

warning

comments to sambo@ida.liu.se

Fig. 7. Screenshot: mapping suggestion.

tion 2.2.2). To generate these segments and segment pairs we first use a string-based approach to detect concepts in the different ontologies with similar names. In the implementation we used exact matching. The concepts in the sub-graphs of the is-a hierarchies of the two ontologies with the matched concepts as roots are then candidate segments and form a candidate segment pair. Among the candidate segment pairs a number of elements (15) of small enough size (maximally 60 concepts in a segment) are retained as segment pairs. As a domain expert or oracle has validated all pairs constructed from the segments, full knowledge is available for the small parts of the ontologies represented by the segments. 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 concepts for which full knowledge is available about the correctness of the mappings between the concepts 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 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). 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 2). 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 non-suggested wrong mappings are assumed to be the common and non-interesting case and are therefore not taken into account in the computation of the similarity. Thus, the Jaccard-similarity is the ratio of the number of suggested correct mappings over the number of suggested mappings plus the number of non-suggested correct mappings.

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.

4.5. User interface

In [13] a cognitive support framework for ontology alignment systems is proposed. The framework was developed using cognitive support theories, a literature review of ontology alignment tools as well as a small observational case study. Different requirements for ontology alignment systems were identified and divided into four conceptual dimensions: analysis and decision making (requirements 1.1-1.4), interaction (requirements 2.1-2.5), analysis and generation (requirements 3.1-3.4), and representation (requirements 4.1-4.7). In this section we discuss the cognitive support of our system using these requirements.

In the analysis and decision making dimension we support the following. In addition to the functionality described earlier, our system has a component for manual ontology alignment where the ontologies are represented as indented trees. In this component the user can select a concept from the first ontology and a concept from the second ontology and manually create a mapping (1.1). It also supports ontology exploration (1.1). The tool provides means for the user to accept/reject mapping suggestions (1.2). Further, the user receives information about the definitions of concepts or relations (1.3). Some information about the context of the concepts and relations is available in the mapping suggestions as well as in the manual alignment component (1.4). In the interaction dimension we support exploration (2.1) and search (2.4) of the ontologies via the manual alignment component. Exploration of potential mappings is supported through the remaining suggestions list (2.2). Further, we support exploration of already verified mappings (2.3) through the history list. The system also supports adding details on verified mappings through the annotation functionality (2.5). In the analysis and generation dimension we support the automatic discovery of mapping suggestions (3.1). The mapping state can be saved and users are allowed to return to a given state (3.3). Potential conflicts arising from adding mappings are detected and the user is notified of potential problems (3.4). Regarding the representation dimension we provide a visual representation of the ontologies using indented trees (4.1). We also provide some information regarding the mappings (4.3) via the annotation functionality. Through our PA-based algorithms we have ways to compute mappable regions (4.4). We provide progress feedback through the different tabs, sessions and the history list (4.6).

There are a number of requirements that are not supported or should be supported in a better way. We do not have a filter strategy for showing, for instance, only mappings with exact names or only mapped concepts in the ontologies (2.4). The current system does not deal with instances and thus does not support the transforming of instances from the source ontology to the target ontology (3.2). The current system detects potential conflicts but does not suggest ways of resolving them (3.4). We have worked on an integrated system for ontology alignment and debugging [18] where (3.4) is the main focus of the work. Although the system provides some information regarding the mapping suggestions (4.2) and mappings (4.3), more information available in the different databases could be presented as well as in a better way. For instance, we do not show explanations on why mapping suggestions were suggested (4.7). Although we have algorithms for computing mappable regions, we do not have a visual presentation of these (4.4). In general, the visualization of the ontologies, mappings and mapping suggestions is subject for future work and different techniques need to be investigated. For instance, indented trees are more organized and familiar to novice users, but a graph visualization may be more controllable and intuitive [14]. Further, we do not identify specific starting points (4.5).

4.6. Running example

A typical alignment process would start a computation session. We do this in a screen such as in Figure 5.

Table 1

Number of correct/wrong mappings that are suggested/not suggested.

	Suggested	Not suggested
Correct	A	C
Wrong	B	D

Table 2

Performance 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)

As there were no previous sessions, we can only click on the 'Start New Session' button. This would lead us to a settings screen such as in Figure 6. Here we need to choose which matchers, combination strategy and filtering strategy (with threshold(s)) to use. As an example, let us choose matchers *TermWN* with weight 2, *n-gram* with weight 1 and *NaiveBayes* with weight 1. This also means that we use the weighted sum combination strategy. Further, we use single threshold filtering with threshold 0.5. We also decide that we want to interrupt the computation after 500 suggestions are computed.

We then start a computation session by clicking the 'Start Computation' button and when 500 mapping suggestions are computed using the previously selected matchers, combination and filtering methods, the session is interrupted and a validation session is started. For the computed mapping suggestions the user is shown a screen as in Figure 7. If a concept in the first ontology appears in several mapping suggestions, these are shown in the same screen. The user can accept mapping suggestions as equivalence or subsumption mappings as well as reject suggestions. In the example in Figure 7 the mapping suggestion would be accepted as an equivalence mapping. After each validation the system removes conflicting mapping suggestions. Using this approach 107 mapping suggestions are removed during the validation session (and thus the user did not need to unnecessarily validate these suggestions). At each time point in a validation session we can also acquire a list of the remaining mapping suggestions as well as our previous validation decisions.

After having validated all mapping suggestions we quit the session. The system then uses the validation decisions to compute recommendations for the settings of the parameters of the alignment algorithms. When we start using the system again, we decide to start a new session. In the settings screen we click on 'Use recommendations from predefined strategies' to view the recommended strategies. We decide to follow the top recommendation of the session-based recommendation strategy that uses the previous validation decisions (but no segments). This recommended strategy uses matchers *n-gram* with weight 2, *TermBasic* with weight 1 and *UMLSM* with weight 1, as well as a double threshold filtering strategy with thresholds 0.6 and 0.7. We decide again to interrupt the computation after 500 suggestions are computed. Only 'new' mapping suggestions will be contained in the 500, i.e., if a mapping suggestion was validated in a previous session it will be considered as handled and therefore not counted.

In the new validation session we decide to interrupt after we have validated 200 suggestions. When we restart the system, in the sessions screen (as in Figure 5) we can select previous sessions or start a new session. We select the validation session that was interrupted and validate the remaining suggestions.

We then continue to use the recommendations by the system using the session-based recommendation strategy that uses the previous validation decisions (but no segments) after every 500 computations and the associated validations. This would lead to three more computation and associated validation sessions. For the first new computation session the recommended strategy uses *n-gram* with weight 1, *TermBasic* with weight 1, *TermWN* with weight 2, *UMLSM* with weight 2, and *NaiveBayes* with weight 2 as well as a double threshold filtering strategy with thresholds 0.3 and 0.5. In the next computation session the recommended strategy is the same as in the previous computation session. In the last computation session the recommended strategy uses *TermBasic* with weight 1, *TermWN* with weight 1, and *UMLSM* with weight 1 as well as a double threshold filtering strategy with thresholds 0.6 and 0.8.

After these sessions there are no more mapping suggestions to validate and we decide to stop the process.

This example run illustrated a rather typical way of using the system. There is, however, additional flexibility. For instance, we can interrupt computation sessions in other ways. We do not need to follow the recommendations by the system or we can use different recommendation strategies at different times. In the example we have used the validation decisions in the recommendations, but we could also use them in the preprocessing or double threshold filtering in the computation sessions by checking the 'use preprocessed data' button in the settings screen.

5. Experiments

We performed several experiments. All experiments show the advantages of *using a session-based system* regarding performance of computation of similarity values, filtering or recommendation. Some of the experiments additionally show how a session-based system can be used for *evaluating strategies* (PA-based algorithms and recommendation algorithms).

As the session-based approach allows for interrupting computation sessions and reusing the computation results from previous sessions, we investigate the influence of using sessions on the efficiency of the ontology alignment system (Section 5.2).

A second main feature of the session-based approach is that it is possible to use validation decisions obtained from previous validation sessions. Therefore, in Section 5.3 we discuss experiments related to one of the issues that can lead to reducing unnecessary user interaction. We investigate the influence using validation decisions from previous sessions for different filtering strategies. The experiments also allow us to evaluate the strategies.

A third main feature of the proposed approach are the recommendation sessions. Little research has been done on this issue, and previous approaches only consider recommendations at the beginning of an alignment session, i.e., without the possibility of using validation decisions from previous sessions. Therefore, in our third set of experiments we investigate different recommendation strategies with and without the session-based approach. Similar to the experiments on filtering, the experiments on recommendation strategies show how the session-based approach, in addition to the actual ontology alignment, also enables evaluation of different strategies. Section 5.4 describes the experiments and provides insights into the algorithms that could not have been obtained (easily) without the session-based approach.

In the remainder of this section we present the experiments set-up as well as describe the findings of the different experiments. For the details of the experiments we refer to the appendix.

5.1. Experiments set-up

We use the OAEI 2014 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 and thus we focus on equivalence mappings in our experiments.

We used the following alignment strategies. We used matchers *n-gram*, *TermBasic*, *TermWN*, *UMLSM* and *NaiveBayes*[9] as introduced in Section 4.2. 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 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 reasons explained in the appendix, in the experiments 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.[10]

For the experiments regarding filtering and recommendation we chose three alignment strategies AS1, AS2, AS3 (see Table 5 in the appendix for details) as

[9] For *NaiveBayes* we generated a corpus of PubMed [45] abstracts. For each concept we used the concept name as a query term for PubMed and retrieved abstracts of documents that contain the query term in their title or abstract using the programming utilities provided by the retrieval system Entrez. 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.

[10] As stated in footnote 2, the best strategy regarding F^c in our implemented system is better than the best system at OAEI Anatomy 2009 and earlier, but better algorithms were reported at OAEI 2010.

a basis for discussion. AS1 is the strategy with best F^c (0.86) and Sim2 (0.75) values. AS2 is an average strategy regarding F^c (0.65) and Sim2 (0.48). AS3 performs poorly for F^c (0.48) and Sim2 (0.32), but has a high R^c value (0.89).

5.2. Computation of Similarity Values

In the first experiment we investigate the influence of using sessions and the similarity values database on the efficiency of the ontology alignment system. 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.[11] However, computation sessions in a session-based approach can take advantage of the fact that previous computation and recommendation sessions already stored similarity values.

Our results show that 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.

5.3. 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 of pre-existing mappings exists. In a session-based approach, however, every validation session generates such sets, which can be used to improve the quality of the mapping suggestions and reduce unnecessary user interaction. Further, the knowledge of the domain expert is taken into account at an early stage. In the following experiments we investigate the influence of sessions and validation decisions for different filtering strategies.

For the strategies AS1, AS2 and AS3 we computed the reduction of the number of mapping suggestions by using the filter strategy that removes mapping suggestions that are in conflict with already validated correct mappings. The main lesson learned is that this strategy is effective and removing such conflicting suggestions should be done as soon as possible. Therefore, in our system we perform the removal after every validation of a correct equivalence mapping and thereby reduce unnecessary user interaction.

Further, as the session-based approach produces validated correct mappings, we can use these in the double threshold filtering approach. We computed the influence of this filtering approach in terms of the total number of mapping suggestions and the number of correct suggestions that are removed by this operation. 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 [29], we experimented with the original ontologies as well as repaired[12] ontologies. The results show that the double threshold 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 the worse the strategy the higher the effect.

5.4. Recommendation Strategies with and without Sessions

The experiments in this section show how recommendation strategies can be used within the session-based framework as well as how well they perform. For these experiments we used Sim2 as recommendation measure.

We set up the experiments such that we could investigate different settings for the recommendation strategies. We investigated in (i) using the session-based approach or using one recommendation session at the beginning of the ontology alignment process. For the session-based approaches we also investigated (ii) the performance of the different recommendation strategies discussed in Section 4.4, (iii) the change of the quality of the recommendation strategies with respect to the validation decisions at hand, (iv) the change of the quality of the overall alignment strategy when recommendation strategies are computed after every interruption and the newly computed recommendation strategy is followed until the next interruption.

For the recommendation algorithm that computes a performance measure for the alignment strategies based on how the strategies perform on the already validated mapping suggestions, we found that the recom-

[11] We note, however, that some systems do cache values. Thanks to Michelle Cheatham for pointing out that the code of several systems shows that caching is used.

[12] See the appendix for how we repaired the ontologies.

mended strategy is always a decent strategy. The best strategy (AS1), however, only appears as a top recommended strategy when we start with a poor strategy (AS3). When we change the recommendation after each session, the recommended strategies are usually good or top strategies. When starting with the best strategy (AS1) the performance is still not so good because of the lack of negative examples (i.e., wrong mapping suggestions), but better in this approach than when the recommendation does not change after every session. We show, however, in the appendix that performance can be improved by generating negative examples when these are not available.

For 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, we found that the lack of negative examples leads to poor results when starting with AS1 or AS2. The recommended strategies have very high recall, but low precision. The results for AS3 show that as the number of processed suggestions increases, the recommended strategy becomes better. This is because the quality of the oracle increases. When we change the recommendation after each session, the recommended strategies in the final recommendations are good or top strategies.

When we do not use sessions we use a 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. The performance of the recommendation algorithm depends on the selected segment pairs.

5.5. Summary of lessions learned

We summarize the lessons learned in Table 3 and discuss them further in the following sections. A first kind of lessons learned relates to the usefulness of the session-based approach. A second kind of lessons learned relates to the algorithms of the actual implemented system. We learned these lessions through experiments with many alignment strategies on the OAEI 2014 Anatomy track ontologies. Although we have used only one pair of ontologies in the experiments, the lessons of the first kind, which is the focus of this paper, are general and are also true for other ontology pairs. By experimenting with other ontology pairs we may, however, learn new lessons about the actual implemented algorithms.

5.5.1. Use of the session-based approach and system

We showed that using the session-based approach leads to alignment quality improvements. As the approach allows for the partial computation and the partial validation of mappings suggestions, validation decisions can be taken into account during the following sessions. The validation decisions represent domain expert knowledge and can be used earlier in the alignment process than in former frameworks. During computation sessions a PA can be used for reducing the search space, which according to the experiments in [27] often leads to an improvement of F^c. As shown in this paper, the use of validation decisions from previous sessions for different kinds of filtering also has a positive effect on F^c. These approaches also reduce unnecessary user interaction.

The session-based approach also supports the recommendation of alignment strategies. As, in general, we do not know which alignment strategies perform well for a particular pair of ontologies, according to our experiments using the recommendations after each session usually leads to better alignments.

Further, during computation and recommendation sessions, computed similarity values are stored in the similarity values database. Using this database in further computation sessions reduces computation times and waiting times for the domain expert.

5.5.2. Lessons about alignment strategies

We also learned some lessions about the actual alignment algorithms. For instance, filtering out suggestions that are in conflict with validation decisions after the locking of sessions is useful and the worse the initial strategy, the more useful this is. Also filtering after the locking of a session using the double threshold filtering method is useful, and the more complete the is-a structure in the ontologies is, the better the results.

The recommendation is important, especially when the initial strategy is not good. It is also clear that the approaches using validation decisions (with and without segment pairs) become better the more suggestions are validated. Further, when using the recommended strategy after each session improves the final result. We also found, that, when too few wrong mapping suggestions are available, we can improve the performance by

Table 3

Lessons learned.

General lessons related to the session-based approach:
Alignment quality improvements.
Domain expert knowledge can be taken into account through the validation decisions.
Reduction of unnecessary user interaction.
Support for recommendations.
Reduction of computation and waiting times.

Lessons related to the implemented system:
Filter suggestions in conflict with validation decisions; use double threshold filtering.
Recommendation is important. The worse the initial approach, the more important recommendation is. More research is needed.

automatically generating wrong mapping suggestions. For the approaches using segment pairs, the experiments show that the choice of the segment pairs influences the recommendation results (which is different from the conclusions of experiments in [52]). Therefore, strategies for choosing segment pairs need to be investigated. In our experiments among the strategies with validation decisions, the strategy with ’validation decisions only’ performed best, but the strategy with ’validation decisions and segment pairs’ may be improved with better segment selection strategies.

6. Related Work

To our knowledge there is no other framework that introduces sessions. Earlier frameworks (e.g., [9,31]) and the systems built according to these frameworks have focused on the generation of mappings suggestions, similar to non-interruptible computation sessions in which validated data usually is not taken into account. Some systems also allow to validate data, similar to non-interruptible validation sessions. As there is no similar framework or system, we briefly address related work regarding the different components and used techniques.

Although there are no other systems that fully implement the session-based approach [19], some systems (e.g., [33,6,13,5,18]) do allow to mimic part of the framework through saving and loading alignments or by having a repository with ontologies and alignments.

The computation of mapping suggestions includes preprocessing, matching, combining and filtering. There are some approaches that reduce the search space by segmenting or partitioning the ontologies and using anchors (concept pairs with high similarity) to connect mappable segments [17,16] or segment similarity [6]. Some approaches use the locality of anchors to reduce the search space [46,53]. In [53] anchors can also be pairs with low similarity values. Another approach uses topic identification and clustering to reduce the search space [4]. The main difference with our approach is that we use validation decisions to partition the ontologies.

For the matching many algorithms have been proposed[13]. As mentioned before, they often implement strategies based on linguistic matching, structure-based strategies, constraint-based approaches, instance-based strategies, strategies that use auxiliary information or a combination of these. The results from OAEI, e.g., [8,1] and evaluation studies such as in [31,2,38] provide some knowledge on the performance of the matchers. In our system we used linguistic matching, instance-based strategies, and strategies that use auxiliary information.

The most commonly used combination strategies are the weighted-sum and the maximum-based approaches. Our system supports these. There are some more advanced combination strategies such as in the schema metamatching framework of [7] and the agent-based method in [49].

Regarding filtering, most systems use single threshold filtering, while we additionally support double threshold filtering. In contrast to most systems, our system can also take into account PAs or validation decisions. Some systems do additional checking of constraints (e.g., [20,16]) thereby adding or removing suggestions.

There are some systems that allow validation of mappings such as SAMBO [31], AlViz [33], COGZ [13] for PROMPT, COMA++ [6], AgreementMaker [5] and AML [42]. None of these systems allow, how-

[13] See e.g., [10,48] and http://ontologymatching.org/.

ever, interruptible sessions. LogMap [21] allows interrupting user interaction. Interrupting user interaction in this case means using heuristics to deal with remaining mapping suggestions. It also allows to pause the user interaction (and save the status) and continue later. There are approaches that try to minimize user interaction. For instance, in [15] minimal mappings are computed for light-weight ontologies and these are presented for validation. AgreementMaker [5] clusters validated mappings and mapping suggestions based on a signature vector and these are shown in a visual analytics panel, thereby dealing with multiple suggestions simultaneously and aiding the user with contextual information. Validation decisions also influence the generation of mapping suggestions. Further, recently, in [40] evaluation measures for user interaction were proposed for which the evaluation can be fully automated.

There are very few recommendation approaches. In [36] it is argued that finding appropriate alignment strategies should be based on knowledge about the strategies and their previous use. As a first step a number of factors (related to input, output, approach, usage, cost and documentation) were identified that are relevant when selecting an alignment strategy. The relevant data is collected by questionnaires. The Analytic Hierarchy Process is used to detect suitable alignment approaches. The results from OAEI and evaluation studies such as in [31,38] could provide useful input data for this approach. In [9], APFEL, a machine learning approach to optimize alignment strategies is proposed. In APFEL a set of feature parameters are declared for the source ontologies, the similarity assessment, and the different matchers, combination and filter algorithms. To generate training data, an existing parametrization is used and mapping suggestions are generated. These suggestions need to be validated by the user. A machine learning approach is then used to learn an optimal parametrization. There are some approaches for tuning the parameters in the ontology alignment systems. The RiMOM [34] and UFOme [44] systems use textual and structural characteristics of the ontologies for the selection of matchers, combinations and filters. The system in [43] uses such characteristics to configure itself in an adaptive way. Falcon-OA [17] includes an approach to tune the thresholds for the matchers.

7. Conclusion

In this paper we presented to our knowledge the first session-based framework that allows a user to interrupt and resume the different stages of the ontology alignment process. The framework enables user involvement, one of the current challenges in ontology alignment, in a natural way. We showed the feasibility of the approach by implementing a session-based version of an existing system. We showed the usefulness of the approach through several experiments with many alignment strategies on the OAEI 2014 Anatomy track ontologies. We also showed 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, including segment selection strategies. A further interesting track is to integrate debugging strategies into the alignment process as in [18]. In a session-based approach debugging can be performed early and thereby increase the quality of the alignment. It would also be useful to develop a software framework that implements the session-based approach and to which existing systems can be plugged in.

Acknowledgments

We acknowledge the financial support of the Swedish e-Science Research Centre (SeRC) as well as the EU FP7 project VALCRI (FP7-IP-608142). We thank Qiang Liu, Muzammil Zareen Khan and Shahab Qadeer for their implementation work on earlier versions of the system. We thank Michelle Cheatham, Daniel Faria and Ernesto Jimenez Ruiz for useful comments to improve the paper.

References

[1] Michelle Cheatham, Zlatan Dragisic, Jerome Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Roger Granada, Valentina Ivanova, Ernesto Jimenez-Ruiz, Patrick Lambrix, Stefano Montanelli, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Alessandro Solimando, Cassia Trojahn, and Ondrej Zamazal. Results of the ontology alignment evaluation initiative 2015. In Pavel Shvaiko, Jerome Euzenat, Ernesto Jimenez-Ruiz, Michelle Cheatham, and Oktie Hassanzadeh,

editors, *Proceedings of the 10th International Workshop on Ontology Matching*, volume 1545 of *CEUR Workshop Proceedings*, pages 60–115, 2015.

[2] Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz, editors, *The Semantic Web - ISWC 2013, 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, volume 8219 of *Lecture Notes in Computer Science*, pages 294–309. Springer, 2013.

[3] Bi Chen, Patrick Lambrix, and He Tan. Structure-based filtering for ontology alignment. In Sumitra Mitra Reddy, editor, *Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 364–369. IEEE Computer Society, 2006.

[4] Agnese Chiatti, Zlatan Dragisic, Tania Cerquitelli, and Patrick Lambrix. Reducing the search space in ontology alignment using clustering techniques and topic identification. In Ken Barker and Jose Manuel Gomez-Perez, editors, *Proceedings of the 8th International Conference on Knowledge Capture*, pages 21:1–21:4. ACM, 2015.

[5] Isabel F Cruz, Cosmin Stroe, and Matteo Palmonari. Interactive user feedback in ontology matching using signature vectors. In Johannes Gehrke, Beng Chin Ooi, Evaggelia Pitoura, and X Sean Wang, editors, *IEEE 28th International Conference on Data Engineering*, pages 1321 – 1324. IEEE Computer Society, 2012.

[6] Hong-Hai Do and Erhard Rahm. Matching large schemas: approaches and evaluation. *Information Systems*, 32(6):857–885, 2007.

[7] Carmel Domshlak, Avigdor Gal, and Haggai Roitman. Rank aggregation for automatic schema matching. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):538–553, 2007.

[8] Zlatan Dragisic, Kai Eckert, Jerome Euzenat, Daniel Faria, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jimenez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Stefano Montanelli, Heiko Paulheim, Dominique Ritze, Pavel Shvaiko, Alessandro Solimando, Cassia Trojahn, Ondrej Zamazal, and Bernardo Cuenca-Grau. Results of the ontology alignment evaluation initiative 2014. In Pavel Shvaiko, Jerome Euzenat, Ming Mao, Ernesto Jimenez-Ruiz, Juanzi Li, and Axel Ngonga, editors, *Proceedings of the 9th International Workshop on Ontology Matching*, volume 1317 of *CEUR Workshop Proceedings*, pages 61–104, 2014.

[9] Marc Ehrig, Steffen Staab, and York Sure. Bootstrapping ontology alignment methods with APFEL. In *The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005. Proceedings*, volume 3729 of *Lecture Notes in Computer Science*, pages 186–200. Springer, 2005.

[10] Jerome Euzenat, Christian Meilicke, Heiner Stuckenschmidt, Pavel Shvaiko, and Cassia Trojahn. Ontology alignment evaluation initiative: Six years of experience. *Journal on Data Semantics*, XV:158–192, 2011.

[11] Jerome Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.

[12] Sean M Falconer and Natalya F Noy. Interactive techniques to support ontology matching. In Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors, *Schema Matching and Mapping*, chapter 2, pages 29–51. Springer, 2011.

[13] Sean M Falconer and Margaret-Anne Storey. A cognitive support framework for ontology mapping. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudre-Mauroux, editors, *The Semantic Web - 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*, volume 4825 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2007.

[14] Bo Fu, Natalya F Noy, and Margaret-Anne Storey. Indented tree or graph? A usability study of ontology visualization techniques in the context of class mapping evaluation. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz, editors, *The Semantic Web - ISWC 2013 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, volume 8218 of *Lecture Notes in Computer Science*, pages 117–134. Springer, 2013.

[15] Fausto Giunchiglia, Vincenzo Maltese, and Aliaksandr Autayeu. Computing minimal mappings. In Pavel Shvaiko, Jerome Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Natasha Noy, and Arnon Rosenthal, editors, *Proceedings of the 4th International Workshop on Ontology Matching*, volume 551 of *CEUR Workshop Proceedings*, pages 37–48, 2009.

[16] Anika Gross, Michael Hartung, Toralf Kirsten, and Erhard Rahm. GOMMA results for oaei 2012. In Pavel Shvaiko, Jerome Euzenat, Anastasios Kementsietsidis, Ming Mao, Natasha Noy, and Heiner Stuckenschmidt, editors, *Proceedings of the 7th International Workshop on Ontology Matching*, volume 946 of *CEUR Workshop Proceedings*, pages 133–140, 2012.

[17] Wei Hu and Yuzhong Qu. Falcon-AO: a practical ontology matching system. *Journal of Web Semantics*, 6(3):237–239, 2008.

[18] Valentina Ivanova and Patrick Lambrix. A unified approach for aligning taxonomies and debugging taxonomies and their alignments. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data - 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2013.

[19] Valentina Ivanova, Patrick Lambrix, and Johan Åberg. Requirements for and evaluation of user support for large-scale ontology alignment. In Fabien Gandon, Marta Sabou, Harald Sack, Claudia d'Amato, Philippe Cudre-Mauroux, and Antoine Zimmermann, editors, *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 – June 4, 2015. Proceedings*, volume 9088 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2015.

[20] Yves R Jean-Mary, E Patrick Shironoshita, and Mansur R Kabuka. Ontology matching with semantic verification. *Journal of Web Semantics*, 7(3):235–251, 2009.

[21] Ernesto Jimenez-Ruiz, Bernardo Cuenca-Grau, Yujiao Zhou, and Ian Horrocks. Large-scale interactive ontology matching: Algorithms and implementation. In Luc De Raedt, Christian

Bessiere, Didier Dubois, Patrick Doherty, and Paolo Frasconi, editors, *20th European Conference on Artificial Intelligence*, pages 444–449. IOS Press, 2012.

[22] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.

[23] Patrick Lambrix and Anna Edberg. Evaluation of ontology merging tools in bioinformatics. In Rus B Altman, A Keith Dunker, Lawrence Hunter, Tiffany A Jung, and Teri E Klein, editors, *Pacific Symposium on Biocomputing 2003*, pages 589–600. World Scientific, 2003.

[24] Patrick Lambrix, Manal Habbouche, and Marta Perez. Evaluation of ontology development tools for bioinformatics. *Bioinformatics*, 19(12):1564–1571, 2003.

[25] Patrick Lambrix and Valentina Ivanova. A unified approach for debugging is-a structure and mappings in networked taxonomies. *Journal of Biomedical Semantics*, 4:10:1–10–19, 2013.

[26] Patrick Lambrix and Rajaram Kaliyaperumal. A session-based approach for aligning large ontologies. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data - 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science*, pages 46–60. Springer, 2013.

[27] Patrick Lambrix and Qiang Liu. Using partial reference alignments to align ontologies. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications - 6th European Semantic Web Conference, ESWC 2009 Heraklion, Crete, Greece, May 31-June 4, 2009 Proceedings*, volume 5554 of *Lecture Notes in Computer Science*, pages 188–202. Springer, 2009.

[28] Patrick Lambrix and Qiang Liu. Debugging the missing is-a structure within taxonomies networked by partial reference alignments. *Data & Knowledge Engineering*, 86:179–205, 2013.

[29] Patrick Lambrix, Qiang Liu, and He Tan. Repairing the missing is-a structure of ontologies. In Asuncion Gomez-Perez, Yong Yu, and Ying Ding, editors, *The Semantic Web - Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009. Proceedings*, volume 5926 of *Lecture Notes in Computer Science*, pages 76–90. Springer, 2009.

[30] Patrick Lambrix, Lena Strömbäck, and He Tan. Information integration in bioinformatics with ontologies and standards. In Francois Bry and Jan Maluszynski, editors, *Semantic Techniques for the Web: The REWERSE perspective*, volume 5500 of *Lecture Notes in Computer Science*, chapter 8, pages 343–376. Springer, 2009.

[31] Patrick Lambrix and He Tan. SAMBO - a system for aligning and merging biomedical ontologies. *Journal of Web Semantics*, 4(3):196–206, 2006.

[32] Patrick Lambrix, Fang Wei-Kleiner, Zlatan Dragisic, and Valentina Ivanova. Repairing missing is-a structure in ontologies is an abductive reasoning problem. In Patrick Lambrix, Guilin Qi, Matthew Horridge, and Bijan Parsia, editors, *Proceedings of the Second International Workshop on Debugging Ontologies and Ontology Mappings*, volume 999 of *CEUR Workshop Proceedings*, pages 33–44, 2013.

[33] Monika Lanzenberger and Jennifer Sampson. AlViz - a tool for visual ontology alignment. In Ebad Banissi, Remo Aslak Burkhard, Anna Ursyn, Jian J Zhang, Mark Bannatyne, Carsten Maple, Andrew J Cowell, Gui Yun Tian, and Ming Hou, editors, *10th International Conference on Information Visualization*, pages 430–440. IEEE Computer Society, 2006.

[34] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. RiMOM: a dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1218–1232, 2009.

[35] Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD thesis, University of Mannheim, 2011.

[36] Malgorzata Mochol, Anja Jentzsch, and Jerome Euzenat. Applying an analytic method for matching approach selection. In Pavel Shvaiko, Jerome Euzenat, Natalya Noy, Heiner Stuckenschmidt, Richard Benjamins, and Michael Uschold, editors, *Proceedings of the 1st International Workshop on Ontology Matching*, volume 225 of *CEUR Workshop Proceedings*, 2006.

[37] DuyHoa Ngo and Zohra Bellahsene. YAM++ results for oaei 2013. In Pavel Shvaiko, Jerome Euzenat, Kavitha Srinivas, Ming Mao, and Ernesto Jimenez-Ruiz, editors, *Proceedings of the 8th International Workshop on Ontology Matching*, volume 1111 of *CEUR Workshop Proceedings*, pages 211–218, 2013.

[38] DuyHoa Ngo, Zohra Bellahsene, and Konstantin Todorov. Opening the black box of ontology matching. In *The Semantic Web: Semantics and Big Data - 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science*, pages 46–60. Springer, 2013.

[39] Natalya F Noy. Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, 33(4):65–70, 2004.

[40] Heiko Paulheim, Sven Hertling, and Dominique Ritze. Towards evaluating interactive ontology matching tools. In *The Semantic Web: Semantics and Big Data - 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2013.

[41] Catia Pesquita, Daniel Faria, Emanuel Santos, and Francisco M Couto. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In Pavel Shvaiko, Jerome Euzenat, Kavitha Srinivas, Ming Mao, and Ernesto Jimenez-Ruiz, editors, *Proceedings of the 8th International Workshop on Ontology Matching*, volume 1111 of *CEUR Workshop Proceedings*, pages 13–24, 2013.

[42] Catia Pesquita, Daniel Faria, Emanuel Santos, Jean-Marc Neefs, and Francisco M Couto. Towards visualizing the alignment of large biomedical ontologies. In Helena Galhardas and Erhard Rahm, editors, *Data Integration in the Life Sciences - 10th International Conference, DILS 2014, Lisbon, Portugal, July 17-18, 2014. Proceedings*, volume 8574 of *Lecture Notes in Computer Science*, pages 104–111. Springer, 2014.

[43] Eric Peuckert, Julian Eberius, and Erhard Rahm. A self-configuring schema matching system. In Johannes Gehrke, Beng Chin Ooi, Evaggelia Pitoura, and X Sean Wang, editors, *28th International Conference on Data Engineering*, pages 306–317. IEEE Computer Society, 2012.

[44] Guiseppe Pirro and Domenico Talia. UFOme: An ontology mapping system with strategy prediction capabilitues. *Data & Knowledge Engineering*, 69:444–471, 2010.

[45] PubMed. http://www.ncbi.nlm.nih.gov/pubmed/.

[46] Md Hanif Seddiqui and Masaki Aono. An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Journal of Web Semantics*, 7(4):344–356, 2009.

[47] Pavel Shvaiko and Jerome Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV:146–171, 2005.

[48] Pavel Shvaiko and Jerome Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, 2013.

[49] Vassilis Spiliopoulos and Gearge A Vouros. Synthesizing ontology alignment methods using the max-sum algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):940–951, 2012.

[50] Unified Medical Language System. http://www.nlm.nih.gov/research/umls/about_umls.html.

[51] He Tan, Vaida Jakoniene, Patrick Lambrix, Johan Aberg, and Nahid Shahmehri. Alignment of biomedical ontologies using life science literature. In Eric G. Bremer, Jörg Hakenberg, Eui-Hong Han, Daniel Berrar, and Werner Dubitzky, editors, *Knowledge Discovery in Life Science Literature - PAKDD 2006 International Workshop, KDLL 2006, Singapore, April 9, 2006. Proceedings*, volume 3886 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2006.

[52] He Tan and Patrick Lambrix. A method for recommending ontology alignment strategies. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudre-Mauroux, editors, *The Semantic Web - 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*, volume 4825 of *Lecture Notes in Computer Science*, pages 494–507. Springer, 2007.

[53] Peng Wang, Yuming Zhou, and Baowen Xu. Matching large ontologies based on reduction anchors. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2243–2348, 2011.

[54] Fang Wei-Kleiner, Zlatan Dragisic, and Patrick Lambrix. Abduction framework for repairing incomplete EL ontologies: Complexity results and algorithms. In Carla E Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1120–1127. AAAI Press, 2014.

[55] WordNet. http://wordnet.princeton.edu/.

8. Appendix: Experiments

8.1. Experiments set-up

As stated before, we use the OAEI 2014 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 and thus we focus on equivalence mappings in our experiments.

We used matchers *n-gram*, *TermBasic*, *TermWN*, *UMLSM* and *NaiveBayes*. 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 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 4 shows the top 10 strategies with respect to Sim2. All these 10 strategies use a weighted-sum combination, double threshold filtering and include *UMLSM* and at least one string matching-based matcher. These strategies have also a 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 filtering 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;0.7, 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 is 259. All 528 strategies have R^w = 1 and 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, 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 regarding filtering (Section 8.3) and recommendation (Section 8.4) we chose three alignment strategies (Table 5) as a basis for discussion. Strategy AS1 uses a weighted sum combination of *TermBasic* with weight 1 and *UMLSM* with weight 1, and double threshold filtering with thresholds 0.4;0.7 (columns 2-4 in Table 5). AS1 generates 1324 mapping suggestions (column 5). AS1 is the strategy with best F^c (0.86) and 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 single threshold filtering with threshold

Table 4

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 5

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

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 single threshold filtering with threshold 0.3. It generates 4061 mapping suggestions.

8.2. Computation of Similarity Values

In the first experiment we investigate the influence of using sessions and the similarity values database on the efficiency of the ontology alignment system. 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.

In Table 6 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.

8.3. Using the Validation Decisions from Previous Sessions for Filtering

In the following experiments we investigate the influence of sessions and validation decisions for filtering.

8.3.1. Filtering using validated correct mappings

Table 7 shows for the strategies AS1, AS2 and AS3 the reduction of the number of mapping 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. The results also suggest that the removal should be done as soon as possible. For instance, for AS3 when we would process 1000 sug-

Table 6

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

gestions without removal, the 156 that would be removed after 500 processed suggestions may actually have been - unnecessarily - validated by the domain expert. 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.

8.3.2. Double threshold filtering using validated correct mappings

In our next 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 8 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 [29], we experimented with the original ontologies (first value) and the repaired[14] 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 the worse the strategy the higher the effect.

8.4. Recommendation Strategies with and without Sessions

The experiments in this section show how recommendation strategies can be used within the session-based framework. Further, we evaluate different recommendation strategies. For these experiments we used Sim2 as recommendation measure. For some of the experiments we also needed to generate segment pairs. We used the method as described in Section 4.4. 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 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.

8.4.1. 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. Tables 9, 10 and 11 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[15], 1000, ..., 4000 suggestions for AS1, AS2 and AS3, respectively. For AS1, AS1 itself does

[14] We repaired (or enriched or completed) the ontologies by adding the missing is-a relations that were detected by logical reasoning on the ontologies and the reference alignment. If A and B belong to one ontology, A' and B' belong to the other ontology, A $\sqsubseteq$ B is derivable in the first ontology, A $\equiv$ A', and B $\equiv$ B', then we should have that A' $\sqsubseteq$ B' is derivable in the second ontology. If this is not the case then A' $\sqsubseteq$ B' is added to the second ontology. More advanced techniques could be used for repairing, e.g., [28,25,54].

[15] 503, because the validation decision for suggestion 500 removes other suggestions.

Table 7
Filter using validated correct mappings.

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

Table 8
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

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 fact that the algorithm for generating consistent groups did not find maximal consistent groups. Therefore, the partitioning allowed for larger mappable groups and thus the double threshold filtering did not filter as much (mainly wrong mapping suggestions) as with a partioning based on a maximal consistent group. For AS2, AS1 does not appear among the top 10 recommendations for all the sessions. The reason for this behavior is the difference between the number of correct mapping suggestions proposed by AS1 and AS2. That is, some of the correct mapping suggestions that are proposed by AS2 will not be proposed by AS1. In this experiment, compared to AS1, the recommended strategies propose 1, 15, 20 and 46 more correct mapping suggestions for 500, 1000, 1500 and 1800 processed suggestions, respectively. We note that the recommended strategy always has an actual $F^c \geq 0.76$ and the strategy which is recommended after 1800 processed suggestions uses a maximum-based combination approach. 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 more wrong mapping suggestions for 503, 1500, 3000, 3500 and 4000 processed suggestions, respectively, which are not suggested by the recommended strategies. The reason for the better performance of the recommended strategy is due to the generated consistent group which allowed for a better partitioning 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).

In general, when using an ontology alignment system with session-based recommendation, a user starts with one alignment strategy and can change strategy based on validations during the alignment process. Therefore, we also performed an experiment where the user starts with AS1, AS2 or AS3, performs sessions in which a maximum of 500 suggestions are processed, and where each new session uses the alignment strategy that is recommended by the recommendation algorithm based on the validation decisions of all previous sessions. The new computation session will only compute 'new' mapping suggestions, i.e., mapping suggestions that were not validated before.

Tables 12, 13 and 14 show results for the recommendation algorithm that uses validation decisions only in which the first sessions were started with the strategies AS1, AS2, and AS3, respectively. The rows in the tables indicate the recommended strategy after each session in the experiments. In the case of AS3 the recommended strategy always has an actual F^c >0.84 which is close to the best strategy F^c (0.86). Similar behavior is observed for AS2, the recommended strategy always has an actual F^c >0.83. In the AS1 case, the recommendation becomes better after session 4. In session 6, both AS1 and AS3 recommended strategies with similar F^c (0.84). Compared to the recommendation results of AS1 shown in Table 9 the quality of recommendation for the AS1 case improves as the number of ses-

Table 9

Session-based recommendation using validation decisions only - AS1.

processed suggestions	matchers	weights	threshold	rec F^c	actual F^c
500	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;1;2;1	0.3;0.6	0.993	0.186
1000	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	2;1;2;1	0.5;0.7	0.992	0.850
1300	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;2;1	0.3;0.7	0.972	0.235

Table 10

Session-based recommendation using validation decisions only - AS2.

processed suggestions	matchers	weights	threshold	rec F^c	actual F^c
500	*n-gram*;*TermBasic*;*UMLSM*	2;1;1	0.6;0.7	0.988	0.834
1000	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;1;2;2;2	0.3;0.5	0.987	0.763
1500	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;1;2;2;2	0.3;0.5	0.981	0.763
1800	*TermBasic*;*UMLSM*;*TermWN*	1;1;1	0.6;0.8	0.98	0.80

Table 11

Session-based recommendation using validation decisions only - AS3.

processed suggestions	matchers	weights	threshold	rec F^c	actual F^c
503	*n-gram*;*TermBasic*;*UMLSM*	1;1;2	0.4;0.8	0.920	0.850
1000	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.950	0.861
1500	*TermBasic*;*UMLSM*;*TermWN*	1;2;1	0.4;0.7	0.940	0.860
2000	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.920	0.861
2500	*TermBasic*;*UMLSM*	1;1	0.4;0.7	0.920	0.861
3000	*UMLSM*;*TermWN*	1;1	0.4;0.7	0.920	0.860
3500	*UMLSM*;*NaiveBayes*;*n-gram* ;*TermBasic*	2;2;1;1	0.3;0.5	0.920	0.860
4000	*n-gram*;*TermBasic*;*UMLSM*	1;1;2	0.5;0.8	0.920	0.860

sions increases. The reason for the better performance is due to the differences in the oracle. In the previous experiment, all the recommendations are done with the mapping suggestions from AS1 which are mostly correct. On the other hand, the validation decisions used in this experiment contain more information about wrong suggestions since the strategy is different for every session.

As the performance for the AS1 case is not good, particularly in the early sessions, and an important cause for this is the lack of negative examples, i.e., wrong mappings suggestions, we investigated whether the performance could be improved by automatically generating negative examples when these are not available or only few are available. We implemented an approach that generates wrong mappings based on mappings validated to be correct. The approach swaps concepts in correct mappings. For instance, if A $\equiv$ A' and B $\equiv$ B' are correct mapping suggestions, then we generate the wrong mapping suggestions A $\equiv$ B' and B $\equiv$ A'. Then we selected the wrong mapping suggestions whose similarity based on edit distance is between 0.45 and 0.65. In this experiment we generated 300 wrong mapping suggestions. Table 15 shows the results for the recommendation when starting the alignment process with AS1. We note that the recommended strategy always has an actual F^c >0.84. This is clearly an improvement compared to the results in Table 12.

8.4.2. *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

Table 12

Using recommended strategy after each session - session-based recommendation using validation decisions only - AS1.

session	matchers	weights	threshold	rec F^c	actual F^c
1	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;1;2;1	0.3;0.6	0.993	0.186
2	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	2;1;1;2;1	0.3;0.6	0.989	0.570
3	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;2;2;2	0.5;0.8	0.989	0.848
4	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;1;2;1	0.3;0.7	0.983	0.635
5	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	2;1;2;1	0.3;0.7	0.964	0.725
6	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;1;1;2;2	0.3;0.6	0.951	0.846

Table 13

Using recommended strategy after each session - session-based recommendation using validation decisions only - AS2.

session	matchers	weights	threshold	rec F^c	actual F^c
1	*n-gram*;*TermBasic*;*UMLSM*	2;1;1	0.6;0.7	0.988	0.834
2	*TermBasic*;*TermWN* ;*UMLSM*;*NaiveBayes*	1;1;1;2	0.3;0.5	0.993	0.856
3	*n-gram*;*TermBasic*;*UMLSM*	1;1;2	0.4;0.8	0.985	0.853
4	*TermBasic*;*UMLSM*;*TermWN*	2;2;1	0.4;0.8	0.972	0.845
5	*UMLSM*;*TermWN*	1;2	0.4;0.8	0.966	0.846
6	*TermWN*;*UMLSM*;*NaiveBayes*;*n-gram*	2;2;1;2	0.4;0.7	0.953	0.838

Table 14

Using recommended strategy after each session - session-based recommendation using validation decisions only - AS3.

session	matchers	weights	threshold	rec F^c	actual F^c
1	*n-gram*;*TermBasic*;*UMLSM*	1;1;2	0.4;0.8	0.920	0.850
2	*TermWN*;*UMLSM*;*NaiveBayes*;*n-gram*	1;2;1;1	0.3;0.7	0.989	0.840
3	*n-gram*;*TermBasic*;*UMLSM*	1;1;2	0.4;0.8	0.978	0.853
4	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;2	0.3;0.5	0.970	0.845
5	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.4;0.8	0.964	0.849
6	*TermWN*;*UMLSM*;*NaiveBayes*;*n-gram*	2;2;2;1	0.3;0.6	0.956	0.845

Table 15

Using recommended strategy after each session - session-based recommendation using validation decisions only - AS1 - automatically generated wrong mapping suggestions.

session	matchers	weights	threshold	rec F^c	actual F^c
1	*UMLSM*	1	0.8	0.980	0.844
2	*UMLSM*;*TermWN*	1;1	0.5	0.950	0.850
3	*n-gram*;*UMLSM*	1;1	0.5	0.943	0.851
4	*n-gram*;*UMLSM*	1;1	0.5	0.938	0.851

on how the strategies perform on the already validated parts of the segment pairs. Tables 16, 17 and 18 show the results for AS1, AS2 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 sessions is one that has very high recall but that also suggests many wrong mappings which the algorithm cannot detect. Similar behavior is observed for AS2, but the oracle used in this case has better information about wrong mapping suggestions than the one which is used in the AS1 case. The recommended strategy for all the sessions has actual F^c = 0.624.

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.

Also for this recommendation stategy we performed an experiment where the user starts with AS1, AS2 or AS3, performs sessions in which a maximum of 500 suggestions are processed, and where each new session uses the alignment strategy that is recommended by the recommendation algorithm based on the validation decisions of all previous sessions. Similarly as before, the new computation session will only compute 'new' mapping suggestions. Tables 19, 20 and 21 show the results for the recommendation algorithm that uses segment pairs and validation decisions. For AS1 and AS3 the algorithm proposed the same strategy for sessions 2 to 4. After session 7, both cases recommended strategies with similar F^c (0.82) values. Even though the best strategy overall (AS1) is not recommended, for the selected segment pairs these are the best strategies and thus the best that the algorithm can propose. We also note that the quality of recommendation for the AS1 case is improved compared with the recommendation results of AS1 in Table 16. For AS2 the algorithm proposed the same strategy for sessions 1 to 3 (F^c = 0.62) and sessions 4 to 8 (F^c = 0.76), the recommendation did not change after session 4. The reason for this behavior is lack of information about correct mappings in the oracle. That is, the segment pairs used in this experiment have a reference alignment of 46 mappings, but the oracle used by the AS2 case has information about only 35 mappings. On the other hand, the oracles used by the AS1 and AS3 cases have information about 42 and 45 mappings, respectively.

8.4.3. 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 (see Table 22).

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.

Table 16
Session-based recommendation using segment pairs and validation decisions - AS1.

processed suggestions	matchers	weights	threshold	rec F^c	actual F^c
500	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
1000	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
1300	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070

Table 17
Session-based recommendation using segment pairs and validation decisions - AS2.

processed suggestions	matchers	weights	threshold	rec F^c	actual F^c
500	*NaiveBayes*;*n-gram*	1;1	0.3;0.5	1	0.624
1000	*NaiveBayes*;*n-gram*	1;1	0.3;0.5	1	0.624
1300	*NaiveBayes*;*n-gram*	1;1	0.3;0.5	1	0.624
1800	*NaiveBayes*;*n-gram*	1;1	0.3;0.5	1	0.624

Table 18
Session-based recommendation using segment pairs and validation decisions - AS3.

processed suggestions	matchers	weights	threshold	rec F^c	actual F^c
503	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
1000	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
1500	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
2000	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
2500	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
3000	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*; *NaiveBayes*	1;1;1;2;1	0.3;0.7	1	0.760
3500	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;1	0.3;0.6	1	0.820
4000	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;1	0.3;0.6	0.990	0.820

Table 19
Using recommended strategy after each session - session-based recommendation using segment pairs and validation decisions - AS1.

session	matchers	weights	threshold	rec F^c	actual F^c
1	*NaiveBayes*;*n-gram*	1;1	0.3;0.8	1	0.070
2	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;2;1;1	0.3;0.6	1	0.143
3	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;2;1;1	0.3;0.6	1	0.143
4	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;2;1;1	0.3;0.6	1	0.143
5	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;2;1;1	0.3;0.6	1	0.143
6	*NaiveBayes*;*n-gram*;*TermBasic*;*TermWN*	1;2;1;1	0.3;0.6	1	0.143
7	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;1	0.5;0.8	1	0.829
8	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;1	0.5;0.8	1	0.829

Table 20

Using recommended strategy after each session - session-based recommendation using segment pairs and validation decisions - AS2.

session	matchers	weights	threshold	rec F^c	actual F^c
1	*n-gram*;*NaiveBayes*	1;1	0.3;0.5	1	0.624
2	*n-gram*;*NaiveBayes*	1;1	0.3;0.5	1	0.624
3	*n-gram*;*NaiveBayes*	1;1	0.3;0.5	1	0.624
4	*n-gram*;*NaiveBayes*	1;1	0.3;0.5	1	0.624
5	*n-gram*;*NaiveBayes*	1;1	0.3;0.5	1	0.624
6	*n-gram*;*NaiveBayes*	1;1	0.3;0.5	1	0.624
7	*n-gram*;*NaiveBayes*	1;1	0.3;0.5	1	0.624
8	*n-gram*;*NaiveBayes*	1;1	0.3;0.5	1	0.624

Table 21

Using recommended strategy after each session - session-based recommendation using segment pairs and validation decisions - AS3.

session	matchers	weights	threshold	rec F^c	actual F^c
1	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
2	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
3	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
4	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.5	1	0.530
5	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*	1;1;1;2	0.3;0.8	1	0.529
6	*n-gram*;*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;1;1;2;1	0.3;0.7	1	0.758
7	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;1	0.3;0.6	1	0.823
8	*TermBasic*;*TermWN*;*UMLSM*;*NaiveBayes*	1;2;2;1	0.3;0.6	1	0.823

Table 22

Session-independent recommendation using segment pairs and oracle.

strategy	F^c	Sim2
AS1	0.93	0.87
AS2	0.68	0.52
AS3	0.64	0.47