Completeness and Optimality in Ontology Alignment Debugging

Jan Noessner[1], Heiner Stuckenschmidt[1],
Christian Meilicke[1], and Mathias Niepert[2]

[1] University of Mannheim, 68163 Mannheim, Germany
firstname@informatik.uni-mannheim.de
[2] University of Washington, Seattle, WA 98195-2350, USA
mniepert@cs.washington.edu

Abstract. The benefit of light-weight reasoning in ontology matching has been recognized by a number of researchers resulting in alignment repair systems such as Alcomo and LogMap. While the general benefit of logical reasoning has been shown in principle, there is no systematic empirical evaluation analyzing (i) the impact of completeness of the reasoning methods and (ii) whether approximate or optimal solutions to the conflict resolution problem have to be preferred. Using standard benchmark data sets, we show that increasing the expressive power does improve the matching results and that optimal resolution methods slightly outperform approximate ones.

Keywords: ontology matching, expressiveness, alignment debugging

1 Introduction

Research in ontology matching has been strongly influenced by earlier results in schema matching [19]. There are several approaches that aim at being universally applicable across ontologies and database schemas by relying on a representation of ontologies and schemas as directed graphs [1]. While various studies have verified the benefit of explicit, logical schema semantics such as description logics and logical reasoning (e.g. [17]), there is only a limited number of approaches that exploit schema semantics to improve matching results in a principled manner. Early approaches exploiting the logical structure of class descriptions were based on specialized similarity measures that take logical operators into account (e.g. [2]). Additional methods avoid structural properties that mimic unwanted reasoning results [6] or require user interaction [18]. More recently, a number of approaches have been proposed that explicitly use ontological reasoning. Meilicke et al., for instance, compute and leverage logical inconsistencies to eliminate conflicts between alignment hypotheses [11]. A related approach was proposed by Jiménez-Ruiz et al. [7]. Additional debugging strategies remove incoherent alignments during a post-processing step [20, 13]. Giunchiglia and colleagues use reasoning over logic-based representations of class labels but solely focus on the problem of matching class hierarchies [4]. Most of these approaches exploit restricted forms of reasoning so as to ensure the scalability to large models. While these approaches demonstrated the benefits of logical reasoning for matching expressive ontologies, there has not been a *systematic* investigation of the impact logical reasoning has on matching results. In particular, it is not obvious whether more expressive reasoning methods provide more benefits than less expressive ones. Furthermore, the impact of applying different strategies for resolving detected logical conflicts, has not been analyzed in details. Within this paper we report about experiments that shed light on both research questions. Another systematic evaluation is provided in [8], where the authors focus on the need of debugging and provide a comparison of two debugging systems, while we focus on completeness and optimality.

The paper is structured as follows. In Section 2 we explain alignment incoherence and introduce the notion of completeness and optimality with respect to alignment debugging. Moreover, we describe three existing debugging systems that we use in our experiments. We discuss the setting of our experiments in Section 3 with a focus on data sets and evaluation metrics. The results of these experiments are presented in Section 4. We close with a discussion in Section 5.

2 Incoherence in Ontology Matching

Ontology Matching is the task of finding correspondences between entities of two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. According to [3], a correspondence between an entity e_1 defined in $\mathcal{O}_1$ and an entity e_2 defined in $\mathcal{O}_2$ is a 4-tuple $\langle e_1, e_2, r, n\rangle$ where r is a semantic relation (such as equivalence), and n is a real-valued confidence value. A set of correspondences is called an alignment. In line with most matching systems and benchmarks, we focus on equivalence correspondences, i.e., $(\langle e_1, e_2, \equiv, n\rangle)$, where the matched entities are either both classes or properties. However, the overall approach can also be applied to any kind of axioms as long as these axioms are supported by the debugging system (e.g., all three systems used in our experiments support also subsumption axioms as correspondences).

An alignment $\mathcal{A}$ can be created by a human expert or by an automated matching system. In both cases, $\mathcal{A}$ might include erroneous correspondences. However, it is reasonable to assume that $\mathcal{O}_1$ and $\mathcal{O}_2$ do not contain erroneous axioms. For that reason, an alignment $\mathcal{A}$ can be interpreted as a set of uncertain, weighted equivalence axioms, while $\mathcal{O}_1 \cup \mathcal{O}_2$ will comprise the certain axioms. Merging $\mathcal{A}$, $\mathcal{O}_1$, and $\mathcal{O}_2$ can then result into an incoherent ontology, i.e. some of the classes of $\mathcal{O}_1$ or $\mathcal{O}_1$ might be unsatisfiable due to the additional information encoded in $\mathcal{A}$. The following example shows an incoherent alignment.

$$\begin{aligned}
\mathcal{O}_1 &= \{\texttt{Jaguar}_1 \sqsubseteq \texttt{Cat}_1, \texttt{Cat}_1 \sqsubseteq \texttt{Animal}_1\},\\
\mathcal{O}_2 &= \{\texttt{Jaguar}_2 \sqsubseteq \texttt{Brand}_2, \texttt{Animal}_2 \sqsubseteq \neg\texttt{Brand}_2\}\\
\mathcal{A} &= \{\langle \texttt{Jaguar}_1 \equiv \texttt{Jaguar}_2, 0.9\rangle, \langle \texttt{Animal}_1 \equiv \texttt{Animal}_2, 0.95\rangle\}
\end{aligned}$$

In this example the classes $\texttt{Jaguar}_1$ and $\texttt{Jaguar}_2$ are unsatisfiable in the merged ontology. There are three possible ways to resolve this incoherence: (1) Discard both correspondences, (2) discard $\langle \texttt{Jaguar}_1 \equiv \texttt{Jaguar}_2, 0.9\rangle$, or (3) discard $\langle \texttt{Animal}_1 \equiv \texttt{Animal}_2, 0.95\rangle$. Obviously, we prefer (2) and (3) over (1). Moreover, it seems to make more sense to remove the correspondence that is less confident, i.e., the most reasonable decision is (2) given no further information is available.

However, with larger matching problems a solution to the debugging problem becomes more complex for two reasons. First, not all conflicts (= subsets of correspondences resulting in incoherence) might be detected. This might be caused by using an incomplete reasoning technique, for example, because only a certain type of axioms are analyzed. Second, the detected conflicts might be overlapping and there are several ways to resolve the incoherence. In such a situation a solution should be preferred that removes as less confidence as possible. We call such a solution an optimal solution and define it as a subset $\Delta \subseteq \mathcal{A}$ such that $\mathcal{A} \setminus \Delta$ is coherent and there exist no other Δ^* such that $\mathcal{A} \setminus \Delta^*$ is coherent and $\sum_{c \in \Delta} conf(c) > \sum_{c \in \Delta^*} conf(c)$. This definition corresponds to the definition of a global optimal diagnosis given in [9].

Note that optimality and completeness are independent characteristics of a debugging system. It is possible to construct a debugging system that is complete in terms of reasoning but cannot guarantee the optimality of the solution, while it is also possible to construct a system that is incomplete and optimal, in the sense that the solution is optimal with respect to all detected conflicts, even though these conflicts are only a subset of all conflicts due to the incompleteness. Note also that the notion of optimality is a technical notion, i.e., an optimal solution might not always be the best solution in terms of precision and recall.

3 Experimental Set-Up

3.1 Datasets

The ontologies we use for the experiments are from the ontology alignment evaluation initiative (OAEI) [5]. We selected the CONFERENCE and the LARGE BIOMED ontologies because these benchmarks are not artificially created (unlike, for instance, the BENCHMARKS dataset), are not focused on a narrow alignment problem (unlike, for instance, the MULTIFARM dataset which is concerned with multilingual ontology matching), and provide coherent reference alignments. Moreover, the size of the LARGE BIOMED ontologies allows us to assess the scalability of the presented approach.

The CONFERENCE dataset consists of 15 ontologies which model the domain of conference organization [21]. The number of classes, properties, and axioms of a particular type of 7 ontologies are listed in Table 1 ordered by increasing expressiveness. Every row in the table, with the exception of the last row, corresponds to one expressiveness level we used for the experiments (see Section 4.1). For the 7 listed ontologies, reference alignments were created for each possible pair, resulting in 21 ontology pairs with a reference alignment.

Since the ontologies in the CONFERENCE dataset are relatively small, we also performed experiments with the LARGE BIOMED ontologies. The corresponding

	cmt	conf.	confof	edas	ekaw	iasted	sigkdd
classes	36	60	38	104	77	140	49
properties	59	64	36	50	33	41	28
subsumption	25	49	33	84	71	132	41
+ disjointness	52	63	76	491	145	133	41
+ domain and range restrictions	149	149	100	543	184	193	73
+ all other $\mathcal{EL}^{++}$ axioms	263	331	293	865	309	505	186
every axiom	318	408	335	903	341	539	193

Table 1. Number of classes, properties, and axioms of the conference ontologies.

data set consists of the Foundational Model of Anatomy (fma)[3], National Cancer Institute Thesaurus (nci)[4], and SNOMED clinical terms[5] ontologies. Semantically rich and with thousands of classes, the problem of aligning these ontologies is one of the computationally most challenging in the OAEI campaign. For the 2013 OAEI campaign, only 12 out of 21 participating system configurations were able to compute results for the three combinations. We used the "small fragment" matching problems of the track. For more details on these data sets we refer the reader to the OAEI track website[6]. The properties of the ontologies are summarized in Table 2.

3.2 Alignment Aggregation

For each of the matching tasks described above, there are several alignments available that have been generated by different matching system. We decided to aggregate these alignment for each matching task in a preprocessing step. Thus, we can work with large input alignments and can avoid an additional subsequent aggregation of the debugging results. We aggregated the results of all matchers participating in the 2013 OAEI campaign. For the conference benchmark, we included all matchers which performed better than the string equality baseline [5]. For the large BioMed benchmark, we included the results of the 6 matchers which were able to compute a solution for every combination [5]. The participants in the large BioMed track were allowed to submit results for different settings of their system. We always used the best results of each system in terms of f-measure.

The method of alignment aggregation resembles the approach described in [9]. For each pair of ontologies, we union the alignments $\mathcal{A}_1, \ldots, \mathcal{A}_i, \ldots, \mathcal{A}_n$ of each matching system i to one alignment $\mathcal{A}$. To that end, we first span the confidence values w of each correspondence $\langle w, a \rangle$ in alignment $\mathcal{A}_i$ to the range of $(0, 1]$. This ensures that the confidence values of the individual matchers are scaled identically. We then compute the aggregated a-priori confidence values for a

[3] `http://sig.biostr.washington.edu/projects/fm/`

[4] `http://ncit.nci.nih.gov/`

[5] `http://www.ihtsdo.org/index.php?id=545`

[6] `http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2013/`

	FMA_{NCI}	$\text{FMA}_{\text{SNOMED}}$	NCI_{FMA}	$\text{NCI}_{\text{SNOMED}}$	$\text{SNOMED}_{\text{FMA}}$	$\text{SNOMED}_{\text{NCI}}$
classes	3696	10157	6488	23958	13412	51128
properties	24	24	63	82	18	51
subsumption	3693	10154	4917	18946	16287	31299
+ disjointness	3732	10196	5022	19099	16287	31299
+ domain and range restrictions	3732	10196	5130	19233	16287	31299
+ all other $\mathcal{EL}^{++}$ axioms	7521	20449	14269	50218	33673	122221
every axiom (without annotations)	7548	20478	15634	54452	47104	122221

Table 2. Number of classes, properties, and deterministic axioms of the large BioMed ontologies. For each ontology there exists two fragments. In case of the fma ontology, for example, one fragment contains the axioms overlapping with the nci ontology and one fragment contains the axioms overlapping with the snomed ontology.

correspondence as the normalized sum of all a-priori confidences of that correspondence. The average size of one alignment for the conference benchmark is 42 ranging from at least 29 to at most 60 correspondences. For the large BioMed benchmark, we obtain 3396 correspondences for the ontology pair nci and fma; 10760 for the pair fma and snomed; and 18842 for snomed and nci.

3.3 Debugging Systems

In our evaluation we present results for the debugging systems ELog, LogMap and Alcomo that we apply on the ontologies and alignments described so far.

- ELog [16] is a reasoner for log-linear description logics, which offers complete reasoning capabilities for EL++. ELog can be used for debugging ontology alignments (details can be found in [15]). Since ELog transforms the debugging problem to finding the MAP state of a Markov Logic Network, it guarantees the optimality of the solution, i.e., the MAP state corresponds to an optimal solution. However, ELog is not complete with respect to the full expressiveness of OWL DL.
- LogMap [7] is a matching system including a component for alignment debugging. In our experiments we report only about applying this component and refer to it, for the sake of simplicity, as LogMap. This component translates the ontologies into a set of Horn clauses and applies the linear Dowling-Gallier algorithm for propositional Horn satisability multiple times for repairing. The algorithm is not optimal and to our knowledge also not complete against the OWL DL profile. LogMap is known to be the most efficient debugging tool currently available (see for example [8]).
- Alcomo [9] has specifically been developed for the purpose of debugging ontology alignments. Alcomo can be used in a setting that ensures the completeness (for OWL DL) and the optimality of the solution. The optimality of the solution is guaranteed by applying an exhaustive search algorithm to check potential solutions. However, this setting is applicable only to small

matching problems. Using a lightweight setting, Alcomo can also be applied to larger matching problems loosing both the features of optimality and completeness.

3.4 Metrics

F-Measure Precision and recall of an alignment $\mathcal{A}$ measure the correctness of $\mathcal{A}$ and the completeness of $\mathcal{A}$, respectively. Both measures are defined with respect to a given reference alignment or gold standard $\mathcal{G}$. The F-measure is the harmonic mean of precision an recall. Precision P, recall R, and F-measure F can be formally defined as

$$P = \frac{|\mathcal{A} \cap \mathcal{G}|}{|\mathcal{A}|}, \quad R = \frac{|\mathcal{A} \cap \mathcal{G}|}{|\mathcal{G}|}, \quad \text{and} \quad F = \frac{2 \cdot P \cdot R}{P + R}.$$

Number of Unsatisfiable Classes The number of unsatisfiable classes is proposed as a quality measure for ontology matching in [10]. It refers to the number of classes that are unsatisfiable in the merged ontology $\mathcal{A} \cup \mathcal{O}_1 \cup \mathcal{O}_2$ where $\mathcal{O}_1$ and $\mathcal{O}_2$ are the matched ontologies and $\mathcal{A}$ is the alignment between them. The smaller the number of unsatisfiable classes the higher the quality of the alignment. We computed the number of unsatisfiable classes with the HermiT [12] reasoner since it is known from previous work [8] that HermiT outperforms other reasoners in the computation of unsatisfiable classes. Unfortunately, we were not able to compute the unsatisfiable classes for the nci and snomed pair under 5 hours and, thus, cannot provide the number of unsatisfiable classes for the large BioMed benchmark.

The conference benchmark experiments were performed on a virtual machine with 8 GB RAM and 2 cores with 2,4 Ghz. The large BioMed experiments were executed on a virtual machine with 60 GB RAM and 2 cores.

4 Experimental Results

4.1 Expressiveness

Within this section we report about experiments that include axiom types with increasing expressiveness. Within these experiments we use the ELog debugging system. For the lowest level of expressiveness, we only include subsumption axioms $A \sqsubseteq B$. For the second level, we add disjointness axioms $A \sqcap B \sqsubseteq \bot$. For the third level, we include domain and range restrictions. Finally, for the most expressive level, we include all axioms representable with the DL $\mathcal{EL}^{++}$. The size of the resulting ontologies is shown in Table 1 and 2 presented in the previous section. The ELog debugging system is complete with respect to each of the resulting matching problems. However, with this approach we simulate different types of debugging systems that are restricted to exploit different levels of expressiveness. For example, on the second level we simulate a debugging

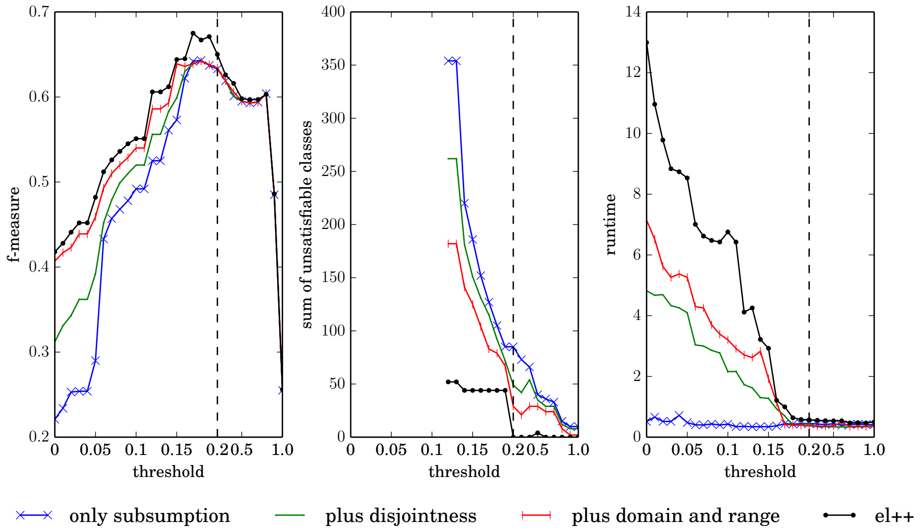

Fig. 1. Results for the CONFERENCE benchmark. With increasing expressiveness, F-measure and runtime (in seconds) are increasing. Contrary, the incoherent classes decrease. This effects are stronger for lower thresholds since more conflicts occur. For thresholds lower than 0.12 the HERMIT reasoner failed in computing the number of incoherent classes. In total, the conference benchmark contains 2.973 classes.

system that bases its reasoning techniques only on the inter-dependencies between subsumption and disjointness axioms. Note that we analyze results for the ontologies in their full expressiveness in the subsequent section.

Figure 1 and Figure 2 depict the results for the various levels of expressiveness and for different thresholds for the CONFERENCE and LARGE BIOMED benchmarks, respectively. The x-axis shows the different thresholds that we applied prior to the debugging step. The results show that the differences between the various levels are less pronounced for lower thresholds. Hence, we put a special emphasis on the threshold areas below 0.2 (for the CONFERENCE benchmark) and below 0.7 (for the LARGE BIOMED benchmark) since results for higher thresholds were nearly identical. Please note that in Figure 1 the stepsize in each chart changes at threshold 0.2 from 0.01 to 0.1 since, beyond that threshold, there are only very few logical conflicts.

We observe a positive correlation between increased expressiveness and F-measure scores. Considering only subsumption axioms results in lower scores compared to the setting with additional disjointness axioms. Even higher F-measures scores are achieved if domain and range restrictions are taken into account. The highest F-measure scores are obtained if we incorporate all $\mathcal{EL}^{++}$ axioms. This holds also true for the choice of a well-suited threshold in the range

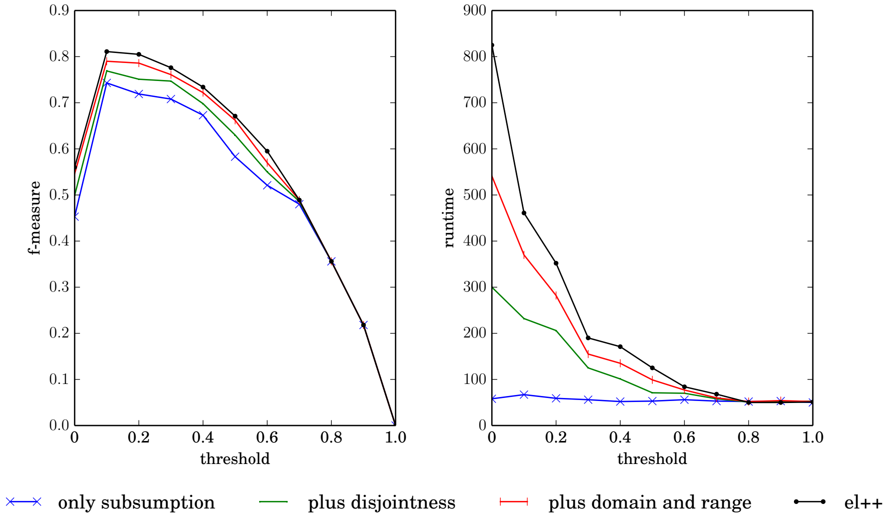

Fig. 2. Results for the large BioMed benchmark. With increasing expressiveness, F-measure scores and running time (in seconds) are increasing. These effects are stronger for lower thresholds since more conflicts occur. We do not provide the number of incoherent classes because the hermiT reasoner did not terminate within 5 hours.

of 0.15 to 0.2 in case of the conference benchmark, where we clearly observe the benefits of exploiting the full expressiveness of $\mathcal{EL}^{++}$.

As expected, the number of unsatisfiable classes (center figure of Figure 1) is higher for settings with decreased expressiveness. For the subsumption only configuration, we observe the highest number of unsatisfiable classes in the final alignment. On the other hand, there are only few unsatisfiable classes for the $\mathcal{EL}^{++}$ setting. Aside from the F-measure results, this is another indication of an improved alignment quality. The reason why we obtain unsatisfiable classes at all for $\mathcal{EL}^{++}$ expressiveness is that the expressiveness of our underlying ontologies is higher than $\mathcal{EL}^{++}$. In case of the large BioMed benchmark the HermiT reasoner was not able to determine the number of unsatisfiable classes within 5 hours. Thus, we do not provide a graphic for this benchmark.

Also as expected, we observe an increase in running time (right figures) when the number of resolved conflicts increases, since runtimes are higher for low thresholds. Furthermore, runtimes also increase with increasing expressiveness. This is in line with our expectation, because a higher level of expressiveness results also in the generation of a more complex optimization problem that needs to be solved when computing the most probable coherent ontology query.

In summary, the results show that the alignment quality increases with an increase in expressiveness. F-measure scores are higher and the number of unsatisfiable classes is lower if expressiveness increases. We can also conclude that

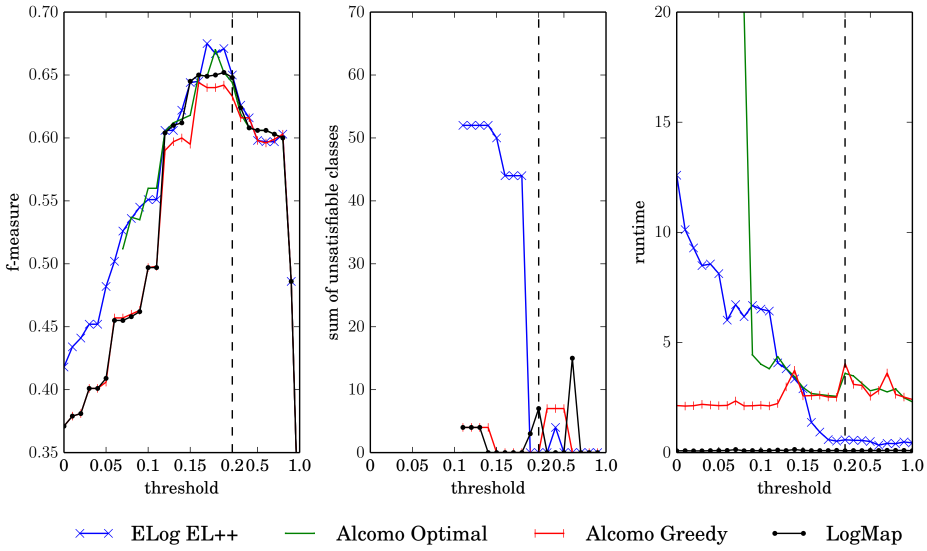

Fig. 3. Results for ELog compared with other approaches on the CONFERENCE benchmark. For lower thresholds, optimal approaches achieve a higher F-measure than approximate approaches but require a longer runtime. The number of unsatisfiable classes is low (1.7% or lower) for all systems. For thresholds lower than 0.12 the HERMIT reasoner failed in computing the number of incoherent classes. In total, the conference benchmark contains 2.973 classes. Runtimes are given in seconds.

a debugging system that is more complete in terms of the supported expressivity will generate better results compared to a less complete system. Runtimes, however, increase with higher expressiveness. This shows a trade-off between runtime and alignment quality depending on the choice of the supported expressiveness.

4.2 Approximate vs. optimal solutions

In this section, we experimentally address the question if optimal algorithms lead to higher quality than approximate algorithms. To that end, we compare the log-linear description logic system ELog and the optimal algorithm of Alcomo against the approximate algorithms of LogMap and Alcomo.[7]

The results for the CONFERENCE and LARGE BioMed benchmark are depicted in Figure 3 and Figure 4, respectively. Again, we focus on the discussion

[7] Alcomo can be executed in different settings. We refer to the setting using the parameters `METHOD_OPTIMAL/REASONING_COMPLETE` as optimal algorithm. We refer to the setting `METHOD_GREEDY/REASONING_EFFICIENT` as approximate algorithm. However, this settings is both incomplete and does not generate an optimal solution.

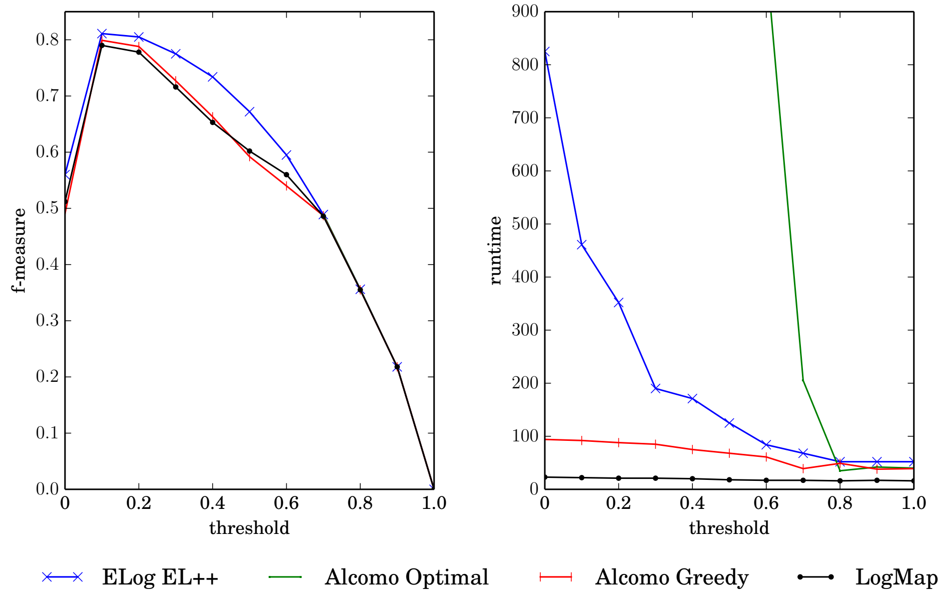

Fig. 4. Results for ELog compared with other approaches on the large BioMed benchmark. For lower thresholds, optimal approaches achieve a higher F-measure than approximate approaches but require a longer runtime. We do not provide the number of incoherent classes because the hermiT reasoner did not terminate within 5 hours. Runtimes are given in seconds.

of results for thresholds below 0.2 (for the conference benchmark) and 0.7 (for the large BioMed benchmark).

The system ELog and the optimal algorithm of Alcomo gains the highest F-measure scores (left figures). The approximate algorithms of Alcomo and LogMap reach lower F-measure scores. The difference in F-measure results between ELog and the optimal algorithm of Alcomo is due to the fact that the associated optimization problems often have more than one solution. Each of this optimal solution has the same objective, i.e. the confidence total of the resulting alignments is the same, but sometimes different F-measure scores. Thus, ELog might choose a different optimum than the optimal algorithm of Alcomo.

ELog has the highest number of unsatisfiable classes (center figure of Figure 3) of all three algorithms. However, having 53 inconsistent classes is only 1.7% compared to the total sum of classes of 2,973. As explained above, ELog is complete only for $\mathcal{EL}^{++}$. Thus, all inconsistencies were caused from axioms which are out of the scope of $\mathcal{EL}^{++}$. The results indicate that the restricted expressivity seems to be less important than the optimality of the solution, since ELog generates at the same time results with the best F-measure.

The approximate algorithms of LogMap and Alcomo are more efficient, especially for lower thresholds. In case of the conference benchmark, ELog outperforms the approximate Alcomo algorithm for thresholds higher than 0.15. Except for the thresholds of 0.11 and 0.12, the exact Alcomo algorithm is slower than ELog and does not terminate within one hour for thresholds below 0.09. For the large BioMed benchmark, the approximate algorithms are faster. For thresholds below 0.7 the exact Alcomo algorithm does not terminate within one hour. LogMap achieves by far the best runtime results, which is also supported by the results reported in [8]. This is (at least partially) caused by incomplete reasoning and non-optimal conflict resolution techniques.

The non-optimal variant of Alcomo and LogMap generate very similar alignments. This becomes obvious when comparing the F-measure scores presented in the left plots of Figure 3 and 4. Obviously, the systems show a similar bevaviour and seem to apply a similar conflict resolution strategy. The same observation can be made for the optimal variant of Alcomo and ELog. Thus, the distinction between optimal and non-optimal algorithms becomes visible in the threshold/F-measure plots, which supports the importance of this distinction.

Overall, we can conclude that optimal systems achieve higher F-measure scores than the approximate algorithms. With respect to runtime, the approximate algorithms are faster than the optimal approaches. In particular LogMap outperforms all other systems. Furthermore, ELog has shorter runtimes than the optimal algorithm of Alcomo. This is remarkable since LogMap and Alcomo are specialized on ontology matching. They leverage the fact that weighted axioms can only occur *between* ontologies and that those axioms are either subsumption or equivalence axioms.

5 Conclusions

Our experiments indicate that an increase in expressiveness leads to an increase in F-measure scores. Furthermore, the comparison of approximate and optimal ontology alignment repairing systems shows that optimal approaches achieve better F-measure scores. However, we observe a trade-off between F-measure and runtime. Runtimes are longer for higher expressiveness and optimal approaches have, on average, longer runtimes than approximate approaches. Thus, we advice users to employ optimal approaches for non-time critical data integration tasks. If real-time ontology alignment is required, we recommend the use of approximate approaches combined with reasoning techniques that might be incomplete.

References

1. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with coma++. In: Proceedings of the 24th International Conference on Management of Data (SIGMOD). pp. 906–908 (2005)
2. Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in OWL-lite. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI). pp. 333–337 (2004)

3. Euzenat, J., Shvaiko, P.: Ontology matching. Springer (2007)
4. Giunchiglia, F., Shvaiko, P.: Semantic matching. Knowledge Eng. Review 18(3), 265–280 (2003)
5. Grau, B.C., Dragisic, Z., Eckert, K., Euzenat, J., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A.O., Lambrix, P., Nikolov, A., Paulheim, H., Ritze, D., Scharffe, F., Shvaiko, P., dos Santos, C.T., Zamazal, O.: Results of the ontology alignment evaluation initiative 2013. In: Proceedings of the 8th Ontology Matching Workshop. pp. 61–100 (2013)
6. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. J. Web Sem. 7(3), 235–251 (2009)
7. Jiménez-Ruiz, E., Grau, B.C.: Logmap: Logic-based and scalable ontology matching. In: Proceedings of the 10th International Semantic Web Conference (ISWC). pp. 273–288 (2011)
8. Jiménez-Ruiz, E., Meilicke, C., Grau, B.C., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: Proceedings of the 26th International Workshop on Description Logics. pp. 246–257 (2013)
9. Meilicke, C.: Alignment incoherence in ontology matching. Ph.D. thesis, University Mannheim (2011)
10. Meilicke, C., Stuckenschmidt, H.: Incoherence as a basis for measuring the quality of ontology mappings. In: Proceedings of the 3rd Ontology Matching Workshop (OM) (2008)
11. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing ontology mappings. In: Proceedings of the 22nd Conference on Artificial Intelligence (AAAI). pp. 1408–1413 (2007)
12. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. Journal of Artificial Intelligence Research 36(1), 165–228 (2009)
13. Ngo, D., Bellahsene, Z., et al.: Yam++-a combination of graph matching and machine learning approach to ontology alignment task. Journal of Web Semantics 16 (2012)
14. Niepert, M., Noessner, J., Stuckenschmidt, H.: Log-linear description logics. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI). pp. 2153–2158 (2011)
15. Noessner, J.: Efficient Maximum A-Posteriori Inference in Markov Logic and Application in Description Logics. Ph.D. thesis, University Mannheim (2014)
16. Noessner, J., Niepert, M.: Elog: A probabilistic reasoner for owl el. In: Proceedings of the 5th Conference on Web Reasoning and Rule Systems (RR). pp. 281–286 (2011)
17. Noy, N., Klein, M.: Ontology evolution: Not the same as schema evolution. Knowledge and Information System 6(4), 428440 (2004)
18. Noy, N.F., Musen, M.A.: Algorithm and tool for automated ontology merging and alignment. In: Proceedings of the 17th National Conference on Artificial Intelligence (2000)
19. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal 10 (2001)
20. Reul, Q., Pan, J.Z.: Kosimap: Use of description logic reasoning to align heterogeneous ontologies. In: 23rd International Workshop on Description Logics DL2010. p. 489. Citeseer (2010)
21. Šváb, O., Svátek, V., Berka, P., Rak, D., Tomášek, P.: Ontofarm: Towards an experimental collection of parallel ontologies. Poster Track of ISWC (2005)