Interaction-Based Ontology Alignment Repair with Expansion and Relaxation

Jérôme Euzenat
Univ. Grenoble Alpes, Grenoble, France
INRIA
Jerome.Euzenat@inria.fr

Abstract

Agents may use ontology alignments to communicate when they represent knowledge with different ontologies: alignments help reclassifying objects from one ontology to the other. These alignments may not be perfectly correct, yet agents have to proceed. They can take advantage of their experience in order to evolve alignments: upon communication failure, they will adapt the alignments to avoid reproducing the same mistake. Such repair experiments had been performed in the framework of networks of ontologies related by alignments. They revealed that, by playing simple interaction games, agents can effectively repair random networks of ontologies. Here we repeat these experiments and, using new measures, show that previous results were underestimated. We introduce new adaptation operators that improve those previously considered. We also allow agents to go beyond the initial operators in two ways: they can generate new correspondences when they discard incorrect ones, and they can provide less precise answers. The combination of these modalities satisfy the following properties: (1) Agents still converge to a state in which no mistake occurs. (2) They achieve results far closer to the correct alignments than previously found. (3) They reach again 100% precision and coherent alignments.

1 Introduction

Agents develop ontologies for representing the world they live in. When confronted to other agents, they use alignments which express the relations holding across entities, e.g., classes or properties, in the ontologies that they use. The result is a network of ontologies related by alignments that agents may use to interoperate. This network is built through interactions between agents that apply corrective actions when communication fails.

The present work aims at eliciting techniques through which agent may adapt these alignments when they find them incorrect. Work has already been performed to understand how this can happen [Euzenat, 2014; Chocron and Schorlemmer, 2016]. We further investigate these by (a) repeating an already performed experiment with updated tools, (b) introducing new similar operators, and (c) introducing new modalities addressing limitations of such operators.

The repetition of existing experiments allows us to correct the reference against which initial results were obtained and to show that agents reach alignments closer to the correct alignments than previously reported. We also highlight that, contrary to previously reported, agents may not achieve full correctness (100% precision). This is due to correct correspondences shadowing incorrect ones.

We introduce new operators and modalities aiming at addressing these problems: new operators correct faulty correspondences by refining them, the expansion modality adds new correspondences when regular operators destroy correspondences, and the relaxation modality enables agents to use shadowed correspondences.

The resulting operators improve the coverage of the alignments through preserving the convergence to 100% success rate, fully correct alignments and a high level of recall (70%).

After reviewing related work (§2) from which the experimental setting (§3) is inspired, we report on repeating one of these experiment with an updated environment (§4). We complement existing operators and show that they improve the results of the previously proposed ones (§5). Then, in order to overcome limitations of the initial random alignments, we introduce and test two modalities allowing agents to introduce random correspondences (§6) and to provide imprecise answers (§7).

2 Related Work

Interacting agents need a way to understand each others to some degree. They can all use the same ontology or preserve heterogeneous ontologies. In the latter case, heterogeneity is often reduced by using alignments between these ontologies [Payne and Tamma, 2014]. However, because matching systems are not always correct or because agents change their ontologies, alignments may become incorrect. There have been various ways to solve this problem independently of agent tasks: gossipping to ensure a global coherence of the networks of ontologies [Aberer *et al.*, 2003], logical repair to enforce consistency [Meilicke, 2011; Jiménez-Ruiz *et al.*, 2013; Santos *et al.*, 2015] or conservativity [Jiménez-Ruiz *et al.*, 2016] constraints. Such approaches have been integrated within agent systems through specific protocols [Aberer *et al.*, 2003; Payne and Tamma, 2014].

Here we are considering approaches in which agents elaborate and repair alignments in the context of other tasks, achieving global repair through local action.

Cultural language evolution [Steels, 2012] showed how a particular culture can be shared by a population of agents through communication. This work offers an experimental methodology: a population of agents has to play randomly an "interaction game" with precise rules. The outcome of the game is clearly identified as a success or a failure. In function of the outcome, agents adapt their language. The state of the system is monitored and especially the success rate which measures the convergence of agents to a stable state. This approach may be applied to language [Steels, 2012] or ontologies [Steels, 1998].

It has been adapted to the evolution of ontology alignments: agents have their own ontologies related by public ontology alignments (network of ontologies). They play games involving the alignments. The games allow them to detect incorrect correspondences through their use and to locally repair the alignment. [Euzenat, 2014] showed that agents are able to repair alignments through playing a simple reclassification game. Several operators were compared and shown to match the capability of logic-based alignment repair systems.

This approach has also been considered in the context of more elaborate games. Interaction-situated semantic alignment [Atencia and Schorlemmer, 2012] considers ontology matching as framed by interaction protocols that agents use to communicate. Agents induce alignments between the different ontologies that they use depending on the success expectation of each correspondence with respect to the protocol. Failing dialogues lead them to revise their expectations and associated correspondences. [Chocron and Schorlemmer, 2016] promotes the combination of the evolutionary approach with classical matching but can be interpreted as a way to repair alignments through their use. This has recently been generalised to less constrained declarative protocols with isomorphic alignments [Chocron and Schorlemmer, 2017].

We consider here very simple games with more elaborate ontologies involving subsumption and disjointness.

3 Experimental Method

The experimental method that is used throughout this paper is directly derived from [Euzenat, 2014] as well as the software which has been used, and extended. Results are thus directly comparable.

The setting is deliberately abstract. This allows to control experiment variables, as our goal is to understand better which factors, in the setting and adaptation operators, influence the properties of the result. These experiments are randomised, which would be difficult with real-world data.

3.1 Setting

Objects in the world are described by a finite set of Boolean features (named a, b, c, etc.). Each object is described by the presence or absence of each feature, e.g., $a \wedge \neg b \wedge \neg c \wedge d$.

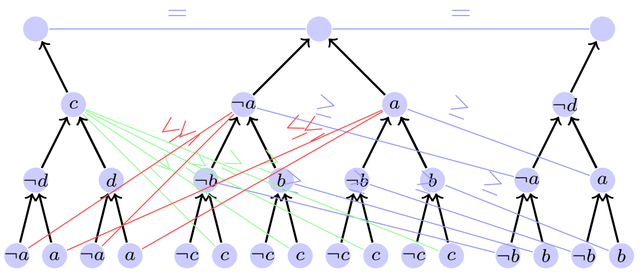

Figure 1: One central ontology and its correct correspondences to two other ontology fragments.

Objects, also called instances, are classified in ontologies which are strict hierarchies. Each class in the hierarchy corresponds to the conjunction of the features of its ancestors. For instance, the bottom-leftmost class in Figure 1 is defined by $c \wedge \neg d \wedge \neg a$. Each level in these hierarchies adds one constraint (positive or negated) related to exactly one feature. This means that classes not in subsumption relation are disjoint. Ontologies are incomplete since they have one level less than the environment has features (d for the middle ontology of Figure 1). Ontologies, expressed in OWL, only use a very simple description logic. The boolean separation, may seem universal and/or artificial, it is simply a minimal language that features subsumption and disjointness.

An alignment is a set of correspondences between two ontologies. A correspondence is an expression like $\langle C, r, C' \rangle$ in which C and C' identify classes of the two ontologies and r is the relation between these classes. We use relation symbols $\leq$, $\geq$ and $=$ denoting subsumption and equivalence relations between classes. Figure 1 shows a fragment of the correct alignments, called reference, between one ontology and two others. Alignments are required to be functional —a class is subsumed by at most one more specific class in each alignment, but one class may subsume several classes— so as to be able to use them to reclassify instances.

3.2 Games

There are as many agents as ontologies (each agent has a different ontology). Agents do not have access to the ontologies of other agents —they know the class names but not their definitions— nor to the reference alignments. They are instead provided with random alignments with the constraint that the topmost classes of each ontology are equivalent and alignments are functional.

In each game, two agents are chosen at random, one agent (A) picks up an instance (i) at random and asks the other one (A') in which class it would classify the instance in its own ontology (O) using the public alignment between O and O'. A' determines which correspondence is applicable and communicates both the correspondence ($\langle C, r, C' \rangle$) and the class (C). A considers the relations between C and the class D in which it would classify i.

If C is compatible with D (C subsumes D, noted $D \sqsubseteq C$) the interaction is considered successful, otherwise (C is disjoint from D, noted $C \perp D$) it is a failure. A then uses an adaptation operator to adapt the alignment (in this case, it only does it in case of failure). Various adaptation operators are discussed in further sections.

3.3 Measures

As agents play games, measures may be applied to the resulting situation (the current network of ontologies) and the evolution of the game. We use the same measures as in the initial experiments:

- **Success rate** [Steels, 2012] characterised by the ratio of success over games played is the main measure.
- **Semantic precision and recall** [Euzenat, 2007] is computed here instead of reporting directly F-measure. They measure the degree of correctness and completeness of the resulting alignments with respect to the known correct reference alignments.[1]
- **Incoherence rate** [Meilicke and Stuckenschmidt, 2008] measures the proportion of incoherent correspondences in alignments taken one by one.
- **Convergence** is the number of games taken to converge in all cases (it is an observed maximum, not an average) when the process converges.

We depart from the initial experiments by using the full reference alignments. It happens that the reference alignments in [Euzenat, 2014] only contained correspondences with leaves. We are now using a new reference alignment with 86 rather than 70 correspondences, in the case of 4 ontologies. As the experiments were always based on instances classified in leaves, this made few differences with respect to the validity of the game, but this led to incorrect F-measure (more precisely: underestimated precision and overestimated recall).

4 Repeated Experiments

We first repeat the second and third experiments of [Euzenat, 2014] concerned respectively with comparing three operators and comparing them to a baseline given by the logical repairers LogMap Repair[2] [Jiménez-Ruiz *et al.*, 2013] and Alcomo [Meilicke, 2011] which restore logical consistency in alignments, by suppressing a minimal set of correspondences. These systems are not in competition as they do not rely on the same type of information: logical repairers have access to all ontologies and alignments but no concrete instance; agents only have access to their ontology and related alignments but can count on the game results.

For repeating the experiments, after implementing new measures and references, we run the *Lazy lavender* platform[3] with 4 agents over 10000 games. All results are the average of 10 runs (not necessarily the same). These are, with new operators over more games (10000 instead of 2000) and improved platform, the same tests as in the initial paper.

4.1 Adaptation Operators

Agents use a specific adaptation operator to solve this problem. The initial experiment compared three operators: delete, replace and add. Assuming that the faulty correspondence $\langle C, r, C'\rangle$ has been crossed by the object from C' to C:

delete simply removes the correspondence;

replace in case r is $=$, then, in addition to delete, the same correspondence with a $\leq$ relation is added ($\langle C, \leq, C'\rangle$);

add a correspondence between C' and the direct superclass C'' of C is added ($\langle C'', \geq, C'\rangle$), in addition to replace.

These operators share two particular properties:

- Safeness: after applying the operator, if the same instance is chosen, the problem would not occur again (maybe a different problem would occur);
- Entailment: each correspondence added by any of the operators was entailed by the removed correspondence.

The properties can be proved a priori. For the first one (safeness), the failure is due to the fact that $D'(i)$, $D' \sqsubseteq C'$, $\langle C, \geq, C'\rangle$ entails $C(i)$, but $C(i)$ conflicts with $D(i)$ and $D \bot C$. Each of these modalities discards $\langle C, \geq, C'\rangle$ and the added correspondences do not permit to deduce $C(i)$. Hence, if it is still entailed, this is for other reasons, that will eventually be found later, e.g., $\langle C, \geq, E'\rangle$ with $C' \sqsubseteq E'$.

For the second property (entailment):

delete: nothing has to be entailed;

replace: $\langle C, =, C'\rangle \models \langle C, \leq, C'\rangle$

add: $\langle C, \geq, C'\rangle, C \sqsubseteq C'' \models \langle C'', \geq, C'\rangle$

4.2 Hypotheses

We expected (H1) to reach the same results as the initial experiment, in particular that (H2) these operators reach 100% precision, since they delete any correspondence that causes failures.

4.3 Results and Discussion

Results are provided in the first two sections of Table 1. The obtained results are in line with those presented before in what concerns convergence, size of the alignments and the respective merits of the operators with respect to F-measure. However, there are three main differences.

The results in terms of F-measure, either concerning LogMap and Alcomo, or the adaptation operators beyond delete are higher than previously. This is due to the new (corrected) reference alignments. This shows that already with 4 agents, agent results are better than logical repairers.

The delete and replace operators have now different F-measure values. This is also due to the new reference.

Contrary to what was expected (H2), agents would not reach 100% precision. It may happen that an incorrect correspondence cannot be detected by the games because it does not cause any failure —such correspondences are shadowed by another correspondence. This is illustrated by Figure 2 (left part): although the red/dashed correspondence is strictly speaking incorrect ($\neg a \wedge \neg b \Rightarrow \neg b \wedge c$ is not true), it happens that whenever it is not true (the class $\neg a \wedge \neg b \wedge \neg c$), the other correspondence (blue/plain) will reclassify instances in the correct class. Hence, the incorrect correspondence does not cause any failure. This is actually like if agents reinvented non-monotonic inheritance: all birds fly ($Bird \rightsquigarrow Fly$), unless they are emus ($Bird \wedge Emu \rightsquigarrow \neg Fly$).

[1] Contrary to classical precision and recall, it is not possible to deduce them from the numbers given hereafter.

[2] We used the version of LogMap made available for OAEI 2016.

[3] http://lazylav.gforge.inria.fr

Network and operator	Size	Success rate	Incoherence degree	Semantic Precision	Semantic F-measure	Semantic Recall	Convergence
reference	86	1.0	0.0	1.0	1.0	1.0	-
initial	54	0.24	0.34	0.11	0.20	0.89	-
Alcomo	28	0.43	0.0	0.21	0.26	0.33	-
LogMap	29	0.51	0.0	0.24	0.26	0.29	-
delete	6	0.99	0.0	1.0	0.13	0.07	445
replace	11	0.99	0.01	0.99	0.21	0.12	1478
add	33	0.98	0.14	0.80	0.52	0.39	1396
refine	20	0.99	0.02	0.96	0.37	0.23	1133
addjoin	23	0.99	0.10	0.84	0.43	0.29	1004
refadd	41	0.99	0.09	0.86	0.62	0.48	1266

Table 1: Measures with the reference and initial network of ontologies, the initial network repaired with Alcomo and LogMap, those obtained by the 3 initial operators (delete, replace, add), and the 3 new operators (refine, addjoin, refadd) [4 agents; 10 runs; 10000 games]. Results of [Euzenat, 2014] are not reported here.

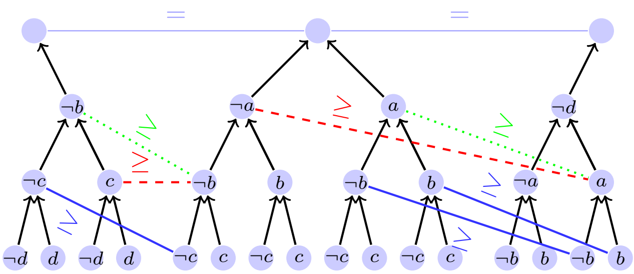

Figure 2: Typical cases of *shadowing* in which agents cannot identify the red/dashed correspondences as incorrect. Indeed, any instance that could highlight the error would be correctly reclassified by the blue/plain correspondences. Thus, the correct correspondences (green/dotted) could not be tested.

As a consequence, not only the result does not reach 100% precision, but it can even be incoherent (as in Figure 2).

5 New Adaptation Operators

Since recall is not very high, we introduced new operators similar to the previous ones expected to provide better results by improving on the previous operators. We first present them and then show how they perform in experiments.

5.1 Adaptation Operators

Three new operators are introduced (with the same faulty correspondence $\langle C, r, C' \rangle$ as above):

addjoin is a variation of add which adds a correspondence between C' and the lowest superclass C'' of C compatible with D ($\langle C'', \geq, C' \rangle$);

refine extends replace by adding a correspondence between C and the subclasses C'' of C' that do not subsume the actual class of the object ($\langle C, \geq, C'' \rangle$);

refadd is the combination of addjoin and refine.

Figure 3 illustrates these operators. It shows two ontology fragments, left and right, with some assertions of disjointness (labelled $\perp$) and subsumption (unlabelled). During a game, the orange triangle instance is drawn and each agent knows to which class it belongs in its ontology (orange arrows). They identify the red correspondence (marked $=$) as erroneous: it leads to the conclusion that this instance belongs to a class which is disjoint from the identified class. delete removes this correspondence. replace replaces it with the subsumption part of the correspondence ($\leq$) that has not been proved incorrect. add (addjoin) will add a ($\geq$) correspondence from a (the common) subsumer of C (and D). The new refine operation will add ($\geq$) correspondences to the subsumees of C' not subsuming D'.

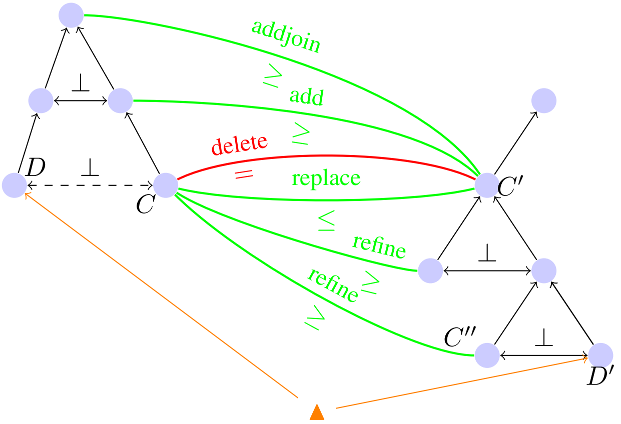

Figure 3: Illustration on two ontology fragments of the effect of the various operators on the correspondence $\langle C, =, C' \rangle$. $\perp$ denotes class disjointness.

These operators still satisfy the properties of safeness and entailment. The argument for safeness is the same as before. Entailment follows from:

addjoin: the argument provided for add is still valid.

refine: $\langle C, \geq, C' \rangle, C'' \sqsubseteq C' \models \langle C, \geq, C'' \rangle$

refadd: inherits from refine and add.

So, the operators are still bounded by the initial random set of alignments.

In addition, to preserve the functionality of alignments, the correspondence is added by refine only if no other correspondence would bring ambiguity.

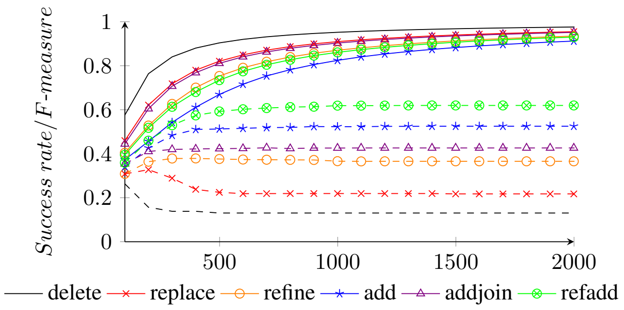

Figure 4: Success rate (plain) and semantic F-measure (dashed) for operators delete/replace/refine/add/addjoin/refadd [4 agents; 10 runs; 10000 games, only displaying the first 2000 games].

refine and refadd require extra steps of interaction to find out the subsumed classes in A''s ontology (alternatively the refine part could be performed by A').

5.2 Hypotheses

Our initial hypotheses are that (H3) refine is better than replace, and (H4) refadd improves over add, addjoin and refine, especially with better recall.

5.3 Results and Discussion

Results are provided in the third section of Table 1. refine loses precision and increases recall when compared to replace and the opposite when compared to add and addjoin). refine is thus found better than replace (H3) and already at least as good as LogMap and Alcomo in terms of F-measure.

Figure 4 shows the evolution over 2000 iterations of the success rate and F-measure of the six presented operators. refadd has an intermediate precision between add-addjoin and replace but, as expected, largely improves both recall. It thus largely improves F-measure (H4) but it is slower to converge than add. The combination of addjoin and refine, refadd is found to have a better recall and F-measure than both.

6 Expansion: Beyond Initial Alignments

The results of these experiments are, as discussed in [Euzenat, 2014], very constrained by the initial random alignments that are provided to the system. Hence, here we propose to expand the set of correspondences and to measure their impact.

6.1 Expansion Modality

We introduce a modality that, when the operators remove a correspondence and are not able to replace it by another correspondence, will add a new random one. This is a modality that can be associated with any operator above.

Agents rely on random correspondences in the sense that there is nothing which guides them towards a particular correspondence. If the correspondence is drawn at random each time, what is observed is that the process does not converge: agents tend to maintain a large number of correspondences (around 100 while the reference alignment has only 86 correspondences) and then they keep on replacing incorrect correspondences by random incorrect ones.

Operator	Size	Success	Inc.	Precision	F-measure	Recall
delete	20	0.98	0.03	0.91	0.35	0.22
replace	35	0.97	0.11	0.84	0.52	0.38
refine	48	0.96	0.16	0.78	0.64	0.55
add	60	0.95	0.21	0.73	0.69	0.66
addjoin	54	0.97	0.17	0.75	0.66	0.59
refadd	66	0.96	0.23	0.71	0.73	0.76

Table 2: Results of all operators with the expansion modality with memory [4 agents; 10 runs; 10000 games].

Hence, the modality rules out some correspondences. First, in order to warrant convergence, the discarded correspondences are recorded and will not be reintroduced since they have been proved incorrect. Moreover, correspondences that the agent can find entailed by its ontology and the current alignment or by a recorded correspondence will not be added.

This modality does not satisfy any more either safeness nor entailment, which were satisfied by the initial operators.

6.2 Hypothesis

Our hypothesis is that (H5) the expansion modality increases recall, and hence overall F-measure, possibly at the expense of a slower convergence.

6.3 Results and Discussion

Results are provided in Table 2. Hypothesis H5 is observed for all operators: they increase recall and decrease precision, increasing F-measure.

Operators still do not reach 100% recall. This is due to two combined phenomena:

- Since instances are always in leaves, as soon as agents found the correct correspondence for a leaf, there is no way to repair the correspondences it shadows (Figure 2).
- By the same phenomenon as before, it does not matter if the shadowed correspondences are correct or not because they are never solicited. Hence, incorrect correspondences may take the place of the correct ones which will never be added (because we enforce functionality).

This is illustrated by Figure 2 (right-hand part): As soon as the agents have found the two correspondences in blue/plain, it is warranted that all instances falling in the two leaf classes will be correctly reclassified. However, agents only play with concrete objects that they classify in the leaves. Hence, whatever correspondence reaches their common ancestors, such as the red/dashed correspondence, will never be tested. In consequence, it will never cause a failure to be repaired.

7 Relaxation: Exploring the Shade

Agents only adapt perfectly to what they have to do, i.e., to succeed in the game. Hence they converge towards 100% success rates. They are not geared to find out nice and consistent theories about the world. Because they reach success in playing the game, incoherence remains.

This is why they should be made more curious. To achieve this, we introduce a specific relaxation modality within the game: Agent A' has the opportunity to not use the most specific correspondence when answering to A, but to chose a

Operator	Size	Success	Precision	F-measure	Recall
delete	6	0.99	1.0	0.13	0.07
replace	11	0.99	1.0	0.20	0.11
refine	19	0.99	1.0	0.33	0.20
add	25	0.98	1.0	0.37	0.23
addjoin	19	0.99	1.0	0.33	0.20
refadd	40	0.98	1.0	0.61	0.44
delete	19	0.98	1.0	0.29	0.17
replace	30	0.97	1.0	0.41	0.26
refine	45	0.96	1.0	0.63	0.46
add	48	0.95	1.0	0.62	0.45
addjoin	46	0.97	1.0	0.60	0.43
refadd	67	0.96	1.0	0.83	0.71

Table 3: Results of all operators with 80% immediate consumption probability [4 agents; 10 runs; 10000 games; im=.8]. The first section uses the raw operators; the second one uses the expansion modality in addition [expand=clever, non redundant]. The incoherence degree is always 0.

more general one. This is alike someone answering a question with a general statement and not the most specific answer she knows (this is Latin music vs. this is Brazilian bossa nova vs. this is a Luiz Bonfá tune).

With relaxation, when encountering a correspondence that applies to the current instance, agents use this correspondence only with a specific probability, called immediate consumption probability, which also applies to the shadowed correspondences. In case there is no shadowed correspondence but the top-most one, agents use the applicable correspondence.

7.1 Hypothesis

It is expected that relaxation enforces the monotonicity of the alignments. Hence, (H6) precision should increase and incoherence decrease.

7.2 Results and Discussion

The same experiments as previously have been performed with the raw operators (as in §4 and 5) and with expansion (§6). We set the immediate consumption probability to 80% (we found this value very close to the optimum, but make no claim about it here).

All operators now achieve 100% precision (Table 3): by having a chance to select the shadowed correspondences, they will be discarded if they are incorrect. This applies to the basic operators (§4 and 5) as well as their expanded version (§6). In consequence, the resulting alignments are always fully coherent (H6).

If relaxation increases precision, it decreases recall with respect to the non relaxing modality, most of the time leading to a lower F-measure. However, the combination of relaxation and expansion with the refadd operator (§5) reaches improved results in both precision and recall leading to a very high F-measure.

One remaining question is why do not agents reach 100% recall? There are two contributing issues: The first one is related to the reverse shadowing problem, illustrated by Figure 5, in which a more general correct but incomplete correspondence shadows its more specific instance. Due to the functionality constraint, the more specific correspondence will never be added. The other issue is that agents do not generate enough correspondences. This is mitigated by the remark that they are able to generate more correspondences than in the reference alignment. However, once converged to a 100% precision configuration, there are no more failure and thus no possibility of completion.

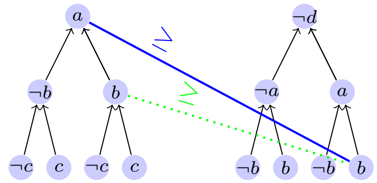

Figure 5: *Reverse shadowing*: the blue/plain correspondence is correct as it asserts that $\neg d \wedge a \wedge b \Rightarrow a$. However, the more specific correspondence $\neg d \wedge a \wedge b \Rightarrow a \wedge b$ (green, dotted), that would entail it, cannot be added because of the functionality constraint.

8 Conclusion

We repeated experiments with agent communication-based alignment repair and found that, contrary to previously reported, agents were not able to fully get rid of incorrect correspondences. We explained why and provided methods to improve existing adaptation operators. First, the refine operator recovers more entailed correspondences than before and the expansion modality allowed to extend alignments beyond initial ones. Then allowing agents to return imprecise answers enable them to test shadowed correspondences. The combination of these three elements leads to full precision and coherence of the resulting alignments and high recall.

With all presented operators, agents converge towards a state in which all communication is finally successful (their goal). As a by-product, applying purely local operators, concerned by one game instance, agents are able to reach global convergence towards correct alignments.

Comparison with logical repairers is not fair any more: They do not have the opportunity to generate and test new correspondences and thus to achieve better recall.

Now that agents are able to generate correspondences by themselves, it becomes possible to make them start from scratch without the provided random correspondences. Determining when and how many correspondences remains a difficult issue.

Finally, with the provided collection of operators (and modalities), it is worth considering how agents, besides selecting correspondences, can select such operators, or how the environment could select agents with specific operators.

Acknowledgments

We thank the developers of Alcomo[4] and LogMap[5] for making their tools available.

[4] http://web.informatik.uni-mannheim.de/alcomo/

[5] https://github.com/ernestojimenezruiz/logmap-matcher

Experiment Material

Experiment records are available under the *Lazy lavender*[3] logbook at entries: [20170214a-NOOR] [20170214b-NOOR] [20170215a-NOOR] [20170215b-NOOR] [20170216-NOOR].

References

[Aberer *et al.*, 2003] Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth. Start making sense: The chatty web approach for global semantic agreements. *Journal of Web Semantics*, 1(1):89–114, 2003.

[Atencia and Schorlemmer, 2012] Manuel Atencia and Marco Schorlemmer. An interaction-based approach to semantic alignment. *Journal of Web Semantics*, 13(1):131–147, 2012.

[Chocron and Schorlemmer, 2016] Paula Chocron and Marco Schorlemmer. Attuning ontology alignments to semantically heterogeneous multi-agent interactions. In *Proc. 22nd ECAI Conference, The Hague (NL)*, pages 871–879, 2016.

[Chocron and Schorlemmer, 2017] Paula Chocron and Marco Schorlemmer. Vocabulary alignment in openly specified interactions. In *Proc. AAMAS, Saõ Paolo (BR)*, pages 1064–1072, 2017.

[Euzenat, 2007] Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, Hyderabad (IN), 2007.

[Euzenat, 2014] Jérôme Euzenat. First experiments in cultural alignment repair (extended version). In *Proc. ESWC 2014 satellite events revised selected papers*, number 8798 in Lecture notes in computer science, pages 115–130, 2014.

[Jiménez-Ruiz *et al.*, 2016] Ernesto Jiménez-Ruiz, Terry Payne, Alessandro Solimando, and Valentina Tamma. Limiting logical violations in ontology alignnment through negotiation. In *Proc. 15th Conferenve on Principles of Knowledge Representation and Reasoning (KR), Cape Town (ZA)*, pages 217–226, 2016.

[Jiménez-Ruiz *et al.*, 2013] Ernesto Jiménez-Ruiz, Christian Meilicke, Bernardo Cuenca Grau, and Ian Horrocks. Evaluating mapping repair systems with large biomedical ontologies. In *Proc. 26th Description logics workshop, Ulm (DE)*, pages 246–257, 2013.

[Meilicke and Stuckenschmidt, 2008] Christian Meilicke and Heiner Stuckenschmidt. Incoherence as a basis for measuring the quality of ontology mappings. In *Proceedings of the 3rd ISWC international workshop on Ontology Matching*, pages 1–12, 2008.

[Meilicke, 2011] Christian Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, Universität Mannheim, 2011.

[Payne and Tamma, 2014] Terry Payne and Valentina Tamma. Negotiating over ontological correspondences with asymmetric and incomplete knowledge. In *Proc. International conference on Autonomous Agents and Multi-Agent Systems (AAMAS) 2014*, pages 517–524, 2014.

[Santos *et al.*, 2015] Emanuel Santos, Daniel Faria, Catia Pesquita, and Francisco Couto. Ontology alignment repair through modularization and confidence-based heuristics. *PLoS ONE*, 10(12):1–19, 2015.

[Steels, 1998] Luc Steels. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous agents and multi-agent systems*, 1(2):169–194, 1998.

[Steels, 2012] Luc Steels, editor. *Experiments in cultural language evolution*. John Benjamins, Amsterdam (NL), 2012.