First experiments in cultural alignment repair

Jérôme Euzenat

INRIA & LIG,
Grenoble, France
Jerome.Euzenat@inria.fr
http://exmo.inria.fr

Abstract. Alignments between ontologies may be established through agents holding such ontologies attempting at communicating and taking appropriate action when communication fails. This approach has the advantage of not assuming that everything should be set correctly before trying to communicate and of being able to overcome failures. We test here the adaptation of this approach to alignment repair, i.e., the improvement of incorrect alignments. For that purpose, we perform a series of experiments in which agents react to mistakes in alignments. The agents only know about their ontologies and alignments with others and they act in a fully decentralised way. We show that such a society of agents is able to converge towards successful communication through improving the objective correctness of alignments. The obtained results are on par with a baseline of a priori alignment repair algorithms.

Keywords: Ontology alignment; alignment repair; cultural knowkedge evolution; agent simulation; coherence; network of ontologies

1 Motivation

The work on cultural evolution applies, an idealised version of, the theory of evolution to culture. Culture is taken here as an intellectual artifact shared among a society. Cultural evolution experiments typically observe a society of agents evolving their culture through a precisely defined protocol. They perform repeatedly and randomly a task, called game, and their evolution is monitored. This protocol aims to experimentally discover the common state that agents may reach and its features. Luc Steels and colleagues have applied it convincingly to the particular artifact of natural language [9].

We aim at applying it to knowledge representation and at investigating some of its properties. A general motivation for this is that it is a plausible model of knowledge transmission. In ontology matching, it would help overcoming the limitations of current ontology matchers by having alignments evolving through their use, increasing the robustness of alignments by making them evolve if the environment evolves.

In this paper, we report our very first experiments in that direction. They consider alignments between ontologies as a cultural artifact that agents may repair while trying to communicate. We hypothesise that it is possible to perform meaningful ontology repair with agents acting locally. The experiments reported here aims at showing that, starting from a random set of ontology alignments, agents can, through a very simple and distributed mechanism, reach a state where (a) communication is always successful, (b) alignments are coherent, and (c) F-measure has been increased. We also compare the obtained result to those of state-of-the-art repair systems.

Related experiments have been made on emerging semantics (semantic gossiping [3, 2]). They involve tracking the communication path and the involved correspondences. By contrast, we use only minimal games with no global knowledge and no knowledge of alignment consistency and coherence from the agents. Our goal is to investigate how agents with relatively little common knowledge (here instances and the interface to their ontologies) can manage to revise networks of ontologies and at what quality.

2 Experimental framework

We present the experimental framework that is used in this paper. Its features have been driven by the wish that experiments be easily reproducible and as simple as possible. We first illustrate the proposed experiment through a simple example (§2.1), before defining precisely the experimental framework (§2.2) following [9].

2.1 Example

Consider an environment populated by objects characterised by three boolean features: color={white|black}, shape={triangle|square} and size={small|large}. This characterises $2^3 = 8$ types of individuals: ■, ▲, □, △, ■, ▲, □, △.

Three agents have their own ontology of what is in the environment. These ontologies, shown in Figure 1, identify the objects partially based on two of these features. Here they are a circular permutation of features: FC (shape, color), CS (color, size) and SF (size, shape).

In addition to their ontologies, agents have access to a set of shared alignments. These alignments comprise equivalence correspondences between their top (all) classes and other correspondences. Initially, these are randomly generated equivalence correspondences. For instance, they may contain the (incorrect) correspondence: SF:small $\equiv$ CS:black.

Agents play a very simple game: a pair of agents a and b are randomly drawn as well as an object of the environment o. Agent a asks agent b the class c (source) to which the object o belongs, then it uses an alignment to establish to which class c' (target) this corresponds in its own ontology. Depending on the respective relation between c and c', a may take the decision to change the alignment.

For instance, if agent CS draws the small-black-triangle (▲) and asks agent SF for its class, this one will answer: small-triangle. The correspondence SF:small $\equiv$ CS:black and the class of ▲ in CS is black-small which is a subclass of CS:black, the result is then a SUCCESS. The fact that the correspondence is not valid is not known to the agents, the only thing that counts is that the result is compatible with their own knowledge.

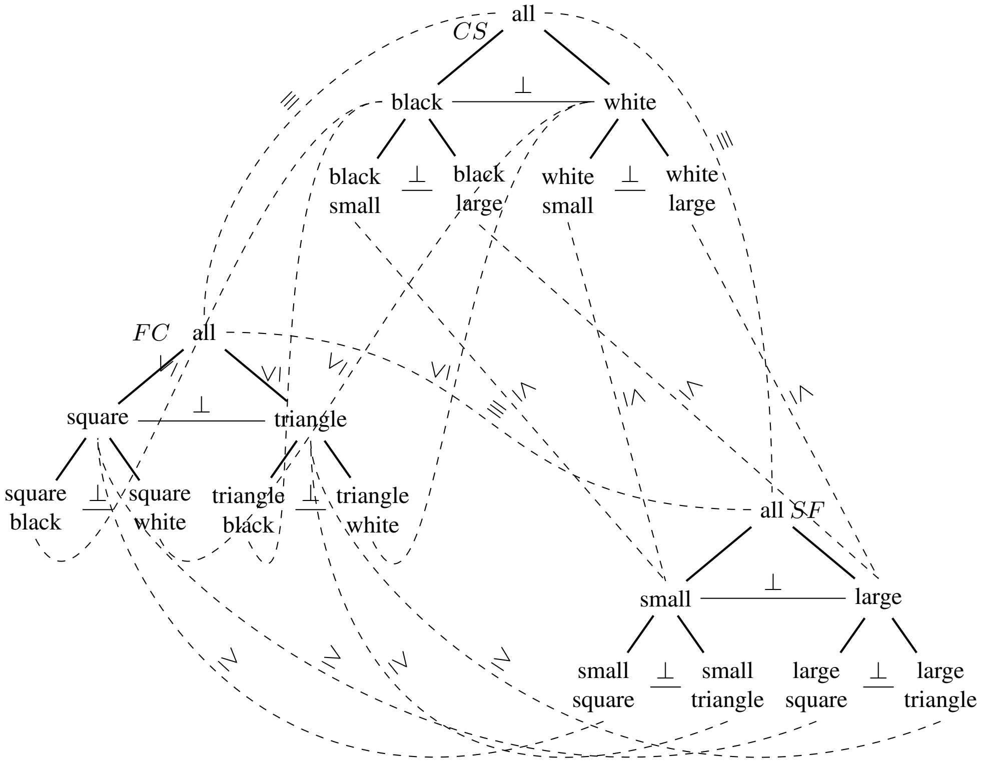

Fig. 1. Example of a generated network of ontologies with the exact reference alignments.

If, on the contrary, the drawn instance is small-white-triangle (△), SF would have made the same answer. This time, the result would be a FAILURE because △ belongs to class CS:white-small which is disjoint from CS:black-small.

How to deal with this failure is a matter of strategy:

delete SF:small $\equiv$ CS:black can be suppressed from the alignment;
replace SF:small $\equiv$ CS:black can be replaced by SF:small $\leq$ CS:black;
add in addition, the weaker correspondence SF:small $\geq$ CS:all can be added to the alignment (but this correspondence is subsumed by SF:all $\equiv$ CS:all).

In the end, it is expected that the shared alignments will improve and that communication will be increasingly successful over time. Successful communication can be observed directly. Alignment quality may be assessed through other indicators: Figure 1 shows (in dotted lines) the correct (or reference) alignments. Reference alignments are not known to the agents but can be automatically generated and used for measuring the quality of the resulting network of ontologies through F-measure.

2.2 Experimental set up

We systematically describe the different aspects of the carried out experiments in the style of [9].
Environment: The environment contains objects which are described by a set of n characteristics (we consider them ordered). Each characteristic can take two possible values which, in this experiment, are considered exclusive.
Population: The experiment uses n agents with as many ontologies. Each agent is assigned one different ontology. In this first setting, each agent will have an ontology based on $n-1$ of these characteristics (each agent will use the first $n-1$ characteristics starting at the agent's rank). The ontology is a simple decision trees of size 2^{n-1} in which each level corresponds to a characteristic and subclasses are disjoint.
Shared network of ontologies: A complete network of $\frac{n\times(n-1)}{2}$ alignments between the ontologies is shared among agents (public). The network is symmetric (the alignment between o and o' is the converse of the alignment between o' and o) and a class is in at most one correspondence per alignment.
Initialisation: In the initial state, each alignment contains equivalence correspondences between the most general classes of both ontologies, plus 2^{n-1} randomly generated equivalence ($\equiv$) correspondences.
Game: A pair of distinct agents $\langle a, b\rangle$ is randomly picked up as well as a set of characteristic values describing an individual (equiprobable). The first agent (a) asks the second one (b) the (most specific) class of its ontology to which the instance belongs ($source$). It uses the alignment between their respective ontologies for finding to which class this corresponds in its own ontology ($target$). This class is compared to the one the instance belongs to in the agent a ontology ($local$).
Success: Full success is obtained if the two classes ($target$ and $local$) are the same. But there are other cases of success:

- $target$ is a super-class of $local$: this is considered successful (this only means that the sets of alignments/ontologies are not precise enough);
- $target$ is a sub-class of $local$: this is not possible here because for each instance, $local$ will be a leaf.

Failure: Failure happens if the two classes are disjoint. In such a case, the agent a will proceed to repair.
Repair: Several types of actions (called modalities) may be undertaken in case of failure:

delete the correspondence is simply discarded from the alignment;
replace if the correspondence is an $\equiv$ correspondence it is replaced by the $\leq$ correspondence from the target class to the source class;
add in addition to the former a new $\leq$ correspondence from the source to a superclass of the target is added. This correspondence was entailed by the initial correspondence, but would not entail the failure.

Success measure: The classical success measure is the rate of successful communication, i.e., communication without failure.
Secondary success measure: Several measures may be used for evaluating the quality of the reached state: consistency, redundancy, discriminability. We use two different measures: the averaged degree of incoherence [7] and the semantic F-measure [4]. Indeed, this setting allows for computing automatically the reference alignment in the network, so we can compute F-measure.
External validation: The obtained result can be compared with that of other repair strategies. We compare the results obtained with those of two directly available repair algorithms: Alcomo [6] and LogMap repair [5].

3 Experiments

We report four series of experiments designed to illustrate how such techniques may work and what are their capabilities

The tests are carried out on societies of at least 4 agents because, in the setting with 3 agents, the delete modality drives the convergence towards trivial alignments (containing only all≡all) and the other modalities do it too often.

All experiments have been run in a dedicated framework that is available from http://lazylav.gforge.inria.fr.

3.1 Convergence

We first test that, in spite of mostly random modalities (random initial alignments, random agents and random instances in each games), the experiments converge towards a uniform success rate.

Four agents are used and the experiment is run 10 times over 2000 games. The evolution of the success rate is compared.

3.2 Modality comparison

The second experiment tests the behaviour of the three repair modalities: delete, replace, add.

Four agents are used and the experiment is run 10 times over 2000 games with each modalities. The results are collected in terms of average success rate and F-measure.

3.3 Baseline comparison

Then the results obtained by the best of these modalities are compared to baseline repairing algorithms in terms of F-measures, coherence and number of correspondences.

The baseline algorithms are Alcomo and LogMap repair. The comparison is made on the basis of success rate, F-measure and the number of correspondences.

LogMap and Alcomo are only taken as a baseline: on the one hand, such algorithms do not have the information that agents may use, on the other hand, agents have no global view of the ontologies and knowledge of consistency or coherence.

3.4 Scale dimension

Finally we observe settings of increasing difficulty by taking the modality providing the best F-measure and applying it to settings with 3, 4, 5 and 6 ontologies.

This still uses 10 runs with the add modality over 10000 games. Results are reported as number of correspondences, F-measure and success rate and compared with the best F-measure of Alcomo and LogMap.

4 Results

Results of the four presented experiments are reported and discussed.

4.1 Convergence

Figure 2 shows the result of our first experiment: 10 runs with a random network as defined above with 4 ontologies. Each curve corresponds to one of the 10 runs over 2000 iterations.

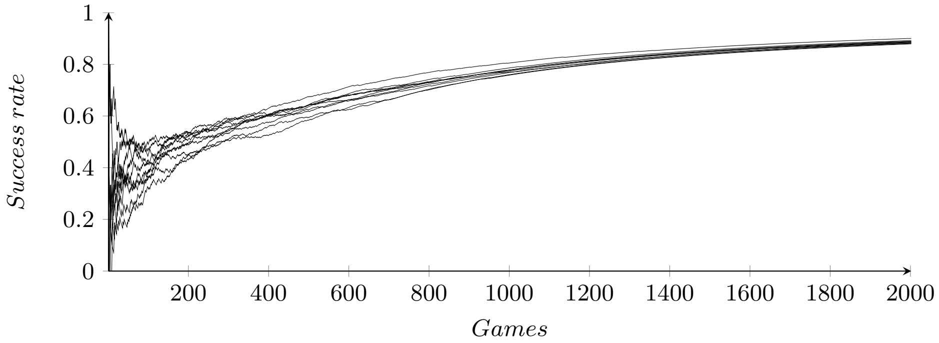

Fig. 2. Ten random runs and their overall success rate, i.e., the proportion of games which were successful so far [mod=add; #agents=4; #games=2000; #runs=1].

Figure 2 shows a remarkable convergence between the runs. After the first 200 games dominated by randomness, they converge assymptotically and at the same pace towards 100%. Indeed, as soon as the network of ontologies has been cleaned up (around 1200 iterations maximum), the rate only grows. It never reaches 1 because of the initial period which contains failures.

From now on, we will still consider 10 runs, but the results will be averaged over these runs.

4.2 Modality comparison

Figure 3 shows the evolution over 2000 iterations of the success rate and F-measure of the three presented modalities.

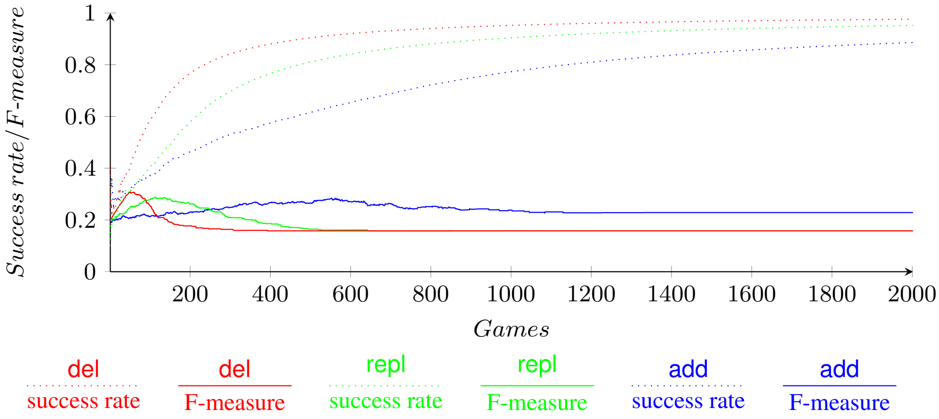

Fig. 3. The average F-measures (dashed) and success rate (plain) with the three different modalities: delete (red), replace (green) and add (blue) [mod=del,repl,add; #agents=4; #games=2000; #runs=10].

delete converges more quickly than replace which converges more quickly than add. This can easily be explained: delete suppresses a cause of problem, replace only suppresses half of it so it may need one further deletion for converging, while add replaces one incorrect correspondence by two correspondences which may be incorrect, so it requires more time to converge.

For the same reason, the success rate is consequently higher. Table 1 shows that for the delete modality, 97.6% success rate corresponds to 48 failure, i.e. 48 deleted correspondences over 54. The 6 remaining correspondences are all≡all correspondences. replace reaches the same result with a 95.2% rate, which corresponds to twice as many failures.

The results of delete and replace modalities are the same: in order to be correct, alignments are reduced to the all≡all correspondences. This is unavoidable for delete (because initial correspondences are equivalences, although, by construction, the correct correspondences are subsumption, so the initial correspondences are incorrect in at least one direction). This is by chance, and because of averaging, for replace.

On the contrary, the add modality has a 88.6% success rate, i.e., 228 failures. This means that on average for each correspondence it has generated 4 alternative correspondences. This is only an average because after 2000 games (and even after 10000 games), there remain more than 12 correspondences.

Contrary to the other modalities, add improves over the initial F-measure.

Table 1 shows that all methods reach full consistency (incoherence rate=0.) from a network of ontologies with 50% incoherence, i.e., half of the correspondences are involved in an inconsistency (or incoherence).

Concerning F-measure, add converges towards a significantly higher value than the two other approaches. With four ontologies, it has a chance to find weaker but more

		Success	Incoherence	Semantic	Syntactic	
Modality	Size	rate	degree	F-measure	F-measure	Convergence
reference	70	-	0.0	1.0	1.0	-
initial	54	-	[0.46-0.49]	0.20	(0.20)	-
delete	6	0.98	0.0	0.16	(0.16)	400
replace	6	0.95	0.0	0.16	(0.16)	1000
add	12.7	0.89	0.0	0.23	(0.16)	1330
Alcomo	25.5	-	0.0	0.26	(0.14)	-
LogMap	36.5	-	0.0	0.26	(0.14)	-

Table 1. Results of the three different modalities compared with Alcomo and LogMap on 10 runs, 4 ontologies and 2000 iterations. Syntactic F-measure has been obtained in an independent but identical evaluation.

correct correspondences. The add strategy is more costly but more effective than the two other strategies.

4.3 Baseline comparison

This experiment exploits the same data as the previous one (§4.2); exploiting those of the next experiment (on 10000 iterations) provides similar results.

Table 1 shows that all three methods are able to restore full coherence and to slightly improve the initial F-measure. Their result is overall comparable but, as can be seen in Figure 4, the agents do not reach the F-measure of logical algorithms.

The agents find half of the correspondences of Alcomo and one third of those of LogMap. This is expected because Alcomo only discards the minimum number of correspondences which bring incoherence, while LogMap weaken them (like the add modality). The agents having more information on what is incorrect, discard more correspondences.

When looking at F-measures, it seems that logical repair strategies can find more than 6 new correspondences which are correct while the add strategy can only find more than 3. This is not true, as shown in Table 1, because we use *semantic* precision and recall [4]. These methods preserve correspondences which are not correct, but which entails correct correspondences. This increases semantic recall and F-measure.

There is a large variation on the results given by the different methods. Out of the same 10 runs, LogMap had the best F-measures 5 times, Alcomo 3 times, and the agents twice. But the largest variation is obtained by the agents with a F-measure ranging from 0.16 to 0.33. Its result is indeed highly dependent on the initial alignment.

4.4 Scale dimension

So far, we concentrated on 4 agents, what happens with a different number of agents? The number of agents does not only determine the number of ontologies. It also determines the number of alignments (quadratic in the number of ontologies), the number of correspondences per alignments and the number of features per instances. This means

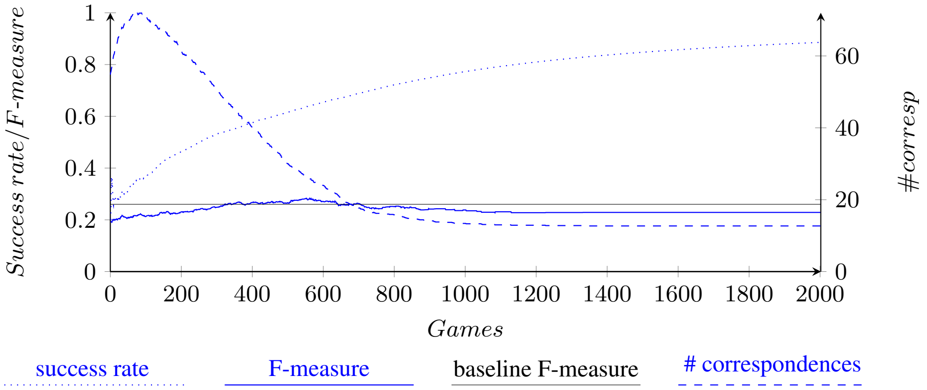

Fig. 4. Average success, F-measure and number of correspondences for the add modality compared to the Alcomo and LogMap F-measure as a baseline [mod=add; #agents=4; #games=2000; #runs=10].

that the more agents are used, the slower is the convergence. So, we played 10000 games in order to have a chance to reach a satisfying level of F-measure.

Figure 5 shows the regular pattern followed by agents: the first phase is random and increases the number of correspondences (due to the add modality). Then, this number slowly decreases. Agents are slower to converge as the problem size increases. This is easily explained: as the correction of the alignment converges, the number of failure-prone games diminishes. Since games are selected at random, the probability to pick up the last configurations (in the end there is only one) becomes lower and lower. The increased number of iterations to converge is directly tied to the largely increased difficulty of the task (number of agents, number of alignments, size of ontologies, characteristics of objects).

This increase is not a measure of the complexity of the approach itself. In fact, it is highly distributed, and it is supposed to be carried out while agents are achieving other tasks (trying to communicate). All the time spend between the two last failures are time of communicative *success*, i.e., agents never had to suffer from the wrong correspondences.

A very simple strategy for improving this would be that agents try to select themselves examples in order to verify the correspondences that they have not already tested.

Table 2 seems to show that, as the complexity of the problem increases, the F-measure of agents is better than that of logical repair mechanisms.

5 Discussion

The relatively low F-measure rate is tied to the type of experiments: agents do not invent any correspondences, they only repair them. Hence, they are constrained by the initial alignment. To this respect, they are on par with logical repair algorithms.

However, they have more information than these repair algorithms. It could then be expected that their results are higher. This is not the case because, when an initial

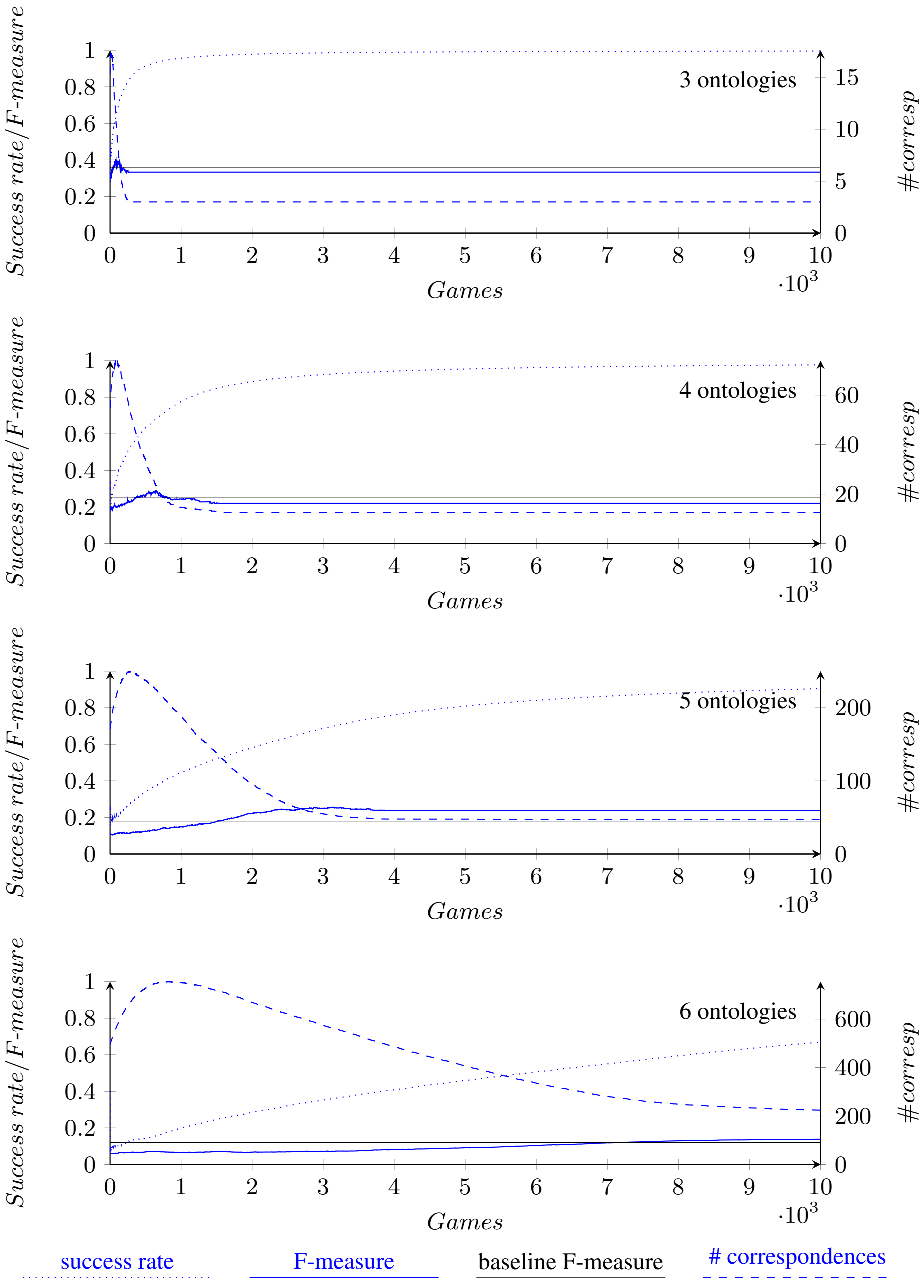

Fig. 5. 10.000 games with 3, 4, 5 and 6 ontologies [mod=add,#agents=3,4,5,6; #games=10000; #runs=10].

	# correpondences					Incoherence				F-measure				
# agents	Reference	Initial	LogMap	Alcomo	Final	Initial	LogMap	Alcomo	Final	Initial	LogMap	Alcomo	Final	Convergence
3	15	15	12	10.3	3	0.31	0.	0.	0.	0.32	0.35	0.36	0.33	300
4	70	54	36.7	28.4	12.4	0.47	0.	0.	0.	0.20	0.24	0.25	0.21	1670
5	250	170	94.7	71.7	47.4	0.58	0.	0.	0.	0.11	0.18	0.17	0.24	5400
6	783	495	234	182	224	0.63	0.	0.	0.	0.06	0.12	0.11	0.14	10.000+

Table 2. Number of correspondences, incoherence rate and F-measure over 10000 games.

correspondence is unrelated to the valid one, agents will simply discard them. They will thus end up with few correspondences with a high precision and low recall.

The state-of-the-art repair algorithms will preserve more correspondences because their only criterion is consistency and coherence: as soon as the alignment is coherent, such algorithms will stop. One could expect a lower precision, but not a higher recall since such algorithms are also tied to the initial alignment.

But because we use semantic precision and recall, it happens that among these erroneous correspondences, some of them entail some valid correspondences (and some invalid ones). This contributes to raise semantic recall.

6 Conclusion

We explored how mechanisms implemented as primitive cultural evolution can be applied to alignment repair. We measured:

- Converging success rate (towards 100% success);
- Coherent alignments (100% coherence);
- F-measures on par with logical repair systems;
- A number of games necessary to repair increasing very fast.

The advantage of this approach are:

- It is totally distributed: agents do not need to have the knowledge of what is an inconsistent or incoherent alignment (only an inconsistent ontology).
- The repair of the network of ontologies is not blind, i.e., restoring inconsistency without knowing if it is likely to be correct, so it also increases F-measure (which is not necessarily the case of other alignment repair strategies [8]).

Yet, this technique does not replace ontology matching nor alignment repair techniques.

7 Perspectives

We concentrated here on alignment repair. However, such a game can perfectly be adapted for matching (creating missing correspondences and revising them on the fly).

In the short term, we would like to adapt this technique in two directions:

- introducing probabilities and using such techniques in order to learn confidence on correspondences that may be used for reasoning [1],
- dealing with alignment composition by propagating instances across agents in the same perspective as the whispering games (propagating classes and see what comes back, setting weights to correspondences) [3].

In the longer term, such techniques do not have to be concentrated on one activity, such as alignment repair. Indeed, they are not problem solving techniques (solving the alignment repair problem). Instead, they are adaptive behaviours, not modifying anything as long as activities are carried out properly, and reacting to improper situations. So, cultural knowledge evolution has to be involved in broader activities, such as information gathering.

8 Acknowledgements

Thanks to Christian Meilicke (Universität Mannheim) and Ernesto Jimenez Ruiz (Oxford university) for making Alcomo and LogMap available and usable. Thanks to an anonymous reviewer for further suggestions.

References

1. Manuel Atencia, Alexander Borgida, Jérôme Euzenat, Chiara Ghidini, and Luciano Serafini. A formal semantics for weighted ontology mappings. In *Proc. 11th International Semantic Web Conference (ISWC)*, volume 7649 of *Lecture notes in computer science*, pages 17–33, Boston (MA US), 2012.
2. Thomas Cerqueus, Sylvie Cazalens, and Philippe Lamarre. Gossiping correspondences to reduce semantic heterogeneity of unstructured P2P systems. In *Proc. 4th International Conference on Data Management in Grid and Peer-to-Peer Systems, Toulouse (FR)*, pages 37–48, 2011.
3. Philippe Cudré-Mauroux. *Emergent Semantics: Interoperability in large-scale decentralized information systems*. EPFL Press, Lausanne (CH), 2008.
4. 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.
5. 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*, 2013.
6. Christian Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, Universität Mannheim, 2011.
7. 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.
8. Catia Pesquita, Daniel Faria, Emanuel Santos, and Francisco Couto. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In *Proc. 8th ISWC ontology matching workshop (OM), Sydney (AU)*, pages 13–24, 2013.
9. Luc Steels, editor. *Experiments in cultural language evolution*. John Benjamins, Amsterdam (NL), 2012.