Knowledge Improvement and Diversity under Interaction-Driven Adaptation of Learned Ontologies

Yasser Bourahla
Univ. Grenoble Alpes, Inria, CNRS,
Grenoble INP, LIG, F-38000 Grenoble
France
Yasser.Bourahla@inria.fr

Manuel Atencia
Univ. Grenoble Alpes, Inria, CNRS,
Grenoble INP, LIG, F-38000 Grenoble
France
Manuel.Atencia@inria.fr

Jérôme Euzenat
Univ. Grenoble Alpes, Inria, CNRS,
Grenoble INP, LIG, F-38000 Grenoble
France
Jerome.Euzenat@inria.fr

ABSTRACT

When agents independently learn knowledge, such as ontologies, about their environment, it may be diverse, incorrect or incomplete. This knowledge heterogeneity could lead agents to disagree, thus hindering their cooperation. Existing approaches usually deal with this interaction problem by relating ontologies, without modifying them, or, on the contrary, by focusing on building common knowledge. Here, we consider agents adapting ontologies learned from the environment in order to agree with each other when cooperating. In this scenario, fundamental questions arise: Do they achieve successful interaction? Can this process improve knowledge correctness? Do all agents end up with the same ontology? To answer these questions, we design a two-stage experiment. First, agents learn to take decisions about the environment by classifying objects and the learned classifiers are turned into ontologies. In the second stage, agents interact with each other to agree on the decisions to take and modify their ontologies accordingly. We show that agents indeed reduce interaction failure, most of the time they improve the accuracy of their knowledge about the environment, and they do not necessarily opt for the same ontology.

KEYWORDS

Ontologies; Multi-agent social simulation; Multi-agent learning; Knowledge diversity

1 INTRODUCTION

In multi-agent systems, agents can rely on ontologies to understand their environment and to interact with each other [22]. When ontologies are learned by agents independently, they may be diverse, incorrect or incomplete. This can cause agent interactions to fail [28]. The origin of these failures differs depending on the interacting agents. Agents may be satisfied by understanding what others mean without necessarily agreeing with them. This may be the case of negotiating agents [15]. For example, a seller agent may consider an object, e.g. a tomato, to be a "vegetable" while a buyer agent would consider it a "fruit". The agents do not agree on what

the object is, but they can still negotiate to reach a price that would satisfy both parties. In contrast, agents may need that others agree with them to carry out tasks together. For example, in a multi-agent cooperation scenario [14], agents that carry goods together will need to agree on what the targeted objects are to achieve their tasks. For that purpose, agents may have to change what they learned from the environment.

However, it is unclear how such changes influence agent knowledge. This raises three fundamental but different questions: Can agents adapt to reach a state with only successful interactions? Is it possible for agents to improve the accuracy of their knowledge about the environment in the process? Do agents preserve the diversity of their knowledge?

To answer these questions, we introduce an experimental framework. It reflects the considered scenarios by two key features. First, agents have to agree with each other about the environment to act successfully. Second, their payoffs depend on the accuracy of their knowledge about the environment. From this we can monitor the evolution of agent performance in their tasks and their knowledge about the environment as they adapt it to agree with each other.

Based on this, a two-stage experiment is designed in which agents first learn to take decisions about objects in the environment and then interact with each other about the decisions to take. In the first stage, from sample objects, agents learn decision classifiers that are converted into ontologies. In the second stage, agents perform tasks which consist in taking decisions in face of various objects. They also interact in random pairs by disclosing the decisions they would take about an object. When agents do not agree on a decision, they adapt their knowledge to correct the cause of the disagreement: the agent considered less skilled, measured through its immediate past ability to accomplish tasks, integrates some of the other's knowledge.

Results show that agents reach a state in which interactions are always successful. Most of the time, they improve their knowledge about the environment but, under specific conditions, knowledge may be forgotten. In addition, they reach this state not necessarily having the same knowledge.

We also applied the same experiment to a real classification dataset and compared its results with a state-of-the-art coordinated learning approach.

The remainder of the paper is organised as follows: in Section 2 we review the related work. Section 3 informally describes the targeted scenario before Section 4 fully specifies the environment, agent abilities and how they interact with each other and perform their tasks. Section 5 presents how the experiments are carried out,

introducing hypotheses, factors and measures. Finally, Section 6 reports and discusses the obtained results.

2 RELATED WORK

Approaches exist that tackle knowledge heterogeneity by finding relationships between different ontologies' entities through ontology matching [13]. These relationships could be precomputed from ontologies or they could be generated dynamically by pairs of agents [20, 26, 29]. In these approaches, agents find relationships between their classes without altering them. This means that agents are only interested in understanding concepts of others through the ones that already exist in their ontologies.

However, an agent's ontology may not include the required concepts to understand another agent's concepts. In this case, agents can extend both their ontologies by negotiating to understand each other's concepts [17]. ANEMONE [28] proposes a communication protocol that allows agents to gradually share parts of their ontologies as they communicate. Agents in ANEMONE can send sample instances in order to help others learn a concept if no satisfactory definition of it could be given. This allows agents to enhance their ontologies and understand the concepts of others. However, ANEMONE does not address the problem when two agents have different understanding of a supposedly same concept. This could be crucial if agents need to agree on a concept to achieve tasks together. This is the case considered here in which agent interactions fail if there is a disagreement about a concept.

Interactions have been exploited as a means for agents to adapt. In interaction-situated semantic alignment [5], agents find alignments between their ontologies based on the success of their interactions. Cultural language evolution [24] showed how agents are able to evolve their language through interaction games. This was also applied to knowledge used to communicate, in the form of ontology alignments [4, 12, 27]. We apply this type of approach to evolve agent ontologies, used to represent the environment, by adapting their classes following interaction failures.

Through such adaptations, agents change what they know about the environment which affects how they perform their classification tasks. A-MAIL [19] is a tool for agents to align learned classifiers, in which agents also can change the classes of objects. Agents aim to coordinate what they learned to be consistent with each other using argumentation, resulting in decentralised inductive learning. Other approaches also rely on argumentation similarly to A-MAIL as a means for agents to reach an agreement on concept meaning [2, 3]. In these approaches, agents deliberately engage in argumentation on what they learned. They are able to generate arguments from the datasets they learned from. In the present paper, agents adapt their knowledge after the learning step and they do not use anymore the data from which they learned but eventually exchange knowledge pieces.

By going through this process, agents end up improving their classifiers and agree on the decisions they take. This is also explored in classifier fusion [10, 23] where multiple classifiers are merged into one model to improve classification performance. This was applied to merging classifiers trained from one data stream [25] as well as from multiple data streams [16, 18]. This is done in a centralised way and results in one classifier while in our case each agent keeps its classifier and adapts it in a decentralised way.

3 INFORMAL SCENARIO DESCRIPTION

To answer the questions raised in the introduction, we consider the following scenario. Agents live in an environment containing various objects. Objects are described by several boolean properties. For example, *canMove*, *hasSharpNails* and *hasEyes* can be the properties describing the objects of an environment. An object *cow* in that environment would have the properties {*canMove*, ¬*hasSharpNails*, *hasEyes*} meaning that it can move, it does not have sharp nails and it has eyes.

Agents take decisions when they encounter objects of the environment. We consider only one type of agent for which each object has one correct decision. For instance, if we consider the decisions *Hunt*, *Leave* and *Collect*, the decision for the object *apple* could be *Collect*, the decision for a *rock* is *Leave* and the decision for *rabbit* is *Hunt*. However, agents do not initially know the correct decisions. Instead, each agent is given a set of sample objects and the corresponding correct decision and learns to take decisions.

When an agent is presented the initial set of samples, it generates a decision tree. For example, two agents a and b learn from two different sets of objects S_1 and S_2, respectively, where $S_1 = \{rock, tiger, rabbit\}$ and $S_2 = \{rabbit, tree\}$. Agent a may learn that objects that cannot move or can move but have sharp nails are to be left and objects that can move and do not have sharp nails are to be hunted. Figure 2 illustrates the corresponding decision tree. Agent b may simply learn that objects that have eyes are hunted and those who do not are left.

After this initial learning phase, agents perform tasks in the environment and interact with each other. Their task is to take decisions about objects of the environment. Taking the right decision provides agents with a payoff correlated to how good its performance was. The payoff could be seen as having food, not being injured, etc.

Each interaction between agents is focused on one object in the environment. Figure 1 illustrates two interactions in which agents a and b interact about the objects "rock" and "lion". The agents disclose the decisions they would take when they encounter the object. Agent a classifies the object "rock" in the decision class *Leave* which is the same class in which agent b classifies it. Since the agents agree on the decisions they take, the interaction is considered successful. However, if the object is a "lion", agent b classifies it in *Hunt* while agent a classifies it in *Leave*, which causes this interaction to fail. This could be seen as two agents who are hunting together but do not agree on whether an object is "huntable" or not. Thus, the hunt will not proceed effectively.

When a failure happens, one of the agents adapts its knowledge to agree with the other agent on the decision to take. This corresponds to learning by communicating. Agents first determine which one of them will change its decision. To do this, they rely on the payoffs received from the environment. The agent having less payoff adapts its knowledge in order to adopt the decision of the other. The intuition behind this is that the payoff denotes success, if not wisdom, and agents tend to imitate the successful ones in an attempt to reach a similar situation.
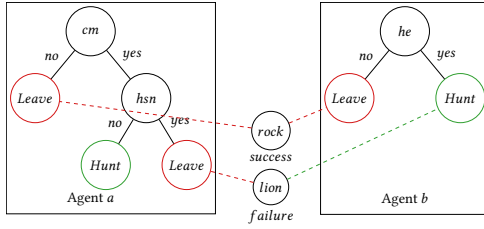
Figure 1: Agents a and b share their decisions for objects "rock" and "lion". For "rock", the decisions are the same so the interaction is successful. For "lion", the decisions are different, hence the interaction is considered a failure. $cm = canMove$, $hsn = hasSharpNails$, $he = hasEyes$.

The initial questions may then be informally reformulated as (1) can agents achieve better communication through these adaptations? (2) Do they improve their knowledge? (3) Do they need to end up with the same ontologies to interact successfully?

4 EXPERIMENTAL FRAMEWORK

To address these questions, we design an experimental framework allowing to deal with scenarios like the previous one. We more precisely introduce the environments and agents before defining the actions that agents perform: ontology learning, interacting and adaptating knowledge.

4.1 Environment

The environment is composed of a set of objects I described by properties from a set $\mathcal{P}$. For the sake of simplicity, the properties are considered binary, i.e. an object either has a property $p \in \mathcal{P}$ or it does not (which is represented by $\neg p$). $\mathcal{P}$ determines the set $\mathbb{I}$ containing one object per possible combination of properties. This means that the size of $\mathbb{I}$ is $2^{|\mathcal{P}|}$. To each of these objects, corresponds only one correct decision in $\mathcal{D} = \{d_1, ..., d_k\}$ given by the function $h^* : I \cup \mathbb{I} \rightarrow \mathcal{D}$.

4.2 Agents

We consider a set A of agents situated in the environment. When an agent encounters an object, it takes a decision about it. Agents know about the possible decisions they can take but not the correct decision about an object. They are also able to see the properties of the objects in the environment: this is shared knowledge between them.

The knowledge of agents is represented as ontologies. Ontologies allow agents to identify objects based on their properties. We use the description logic $\mathcal{ALC}$ [6] to express ontologies. We denote that a class C is subsumed by another class D by $C \sqsubseteq D$, equivalent to D by $C \equiv D$ or disjoint from D by $C \oplus D$. $\top$ and $\bot$ are the top and bottom classes representing the class containing all individuals and the empty class, respectively. From the classes C and D, the union ($C \sqcup D$), the intersection ($C \sqcap D$) and the negation ($\neg C$) can be formed. Constraint on properties may be $\exists p.C$ (objects having at least a value of property p in class C) or $\forall p.C$ (objects having all values of property p in class C). We restrict the use of $\mathcal{ALC}$ such that agents only use $\exists p.\top$ (objects having a value for property p)

and $\forall p.\bot$ (objects having no value for property p) that we note as p and $\neg p$, respectively. The right-hand side of Figure 2 shows the ontology corresponding to the decision tree of agent a in Figure 1.

4.3 Ontology Learning

Initially, each agent is provided with a possibly different training set. This training set contains a subset of objects of I, each with a different property combination, associated with the corresponding correct decision (labelled samples). The proportion r of objects in the sample with respect to $\mathbb{I}$ is called the training ratio. From the training set each agent learns a decision tree classifier, using the ID3 algorithm [21].

In a decision tree, each node corresponds to a test on a property. Each sub-branch of a node corresponds to one outcome of the test. Leaf nodes are associated to decisions. Each object satisfies the tests leading to only one leaf node from the root. It is classified in the decision associated to that leaf node.

Nodes can be viewed as classes of objects and each child node corresponds to a subclass restricting one property to a value. Based on this principle [9], the decision tree is transformed into an ontology in $\mathcal{ALC}$ (see Figure 2). The ontologies obtained from decision trees are private to each agent.

Agents know about the possible decisions. This can be thought of as if they had access to a common external ontology O^* of the possible decisions without knowing their definitions. This ontology contains for each decision d_i a class D_i, and every two different decision classes D_i and D_j are disjoint. Agents can express that objects of class C in its own ontology correspond to a decision d_i by adding the correspondence $\langle C, \sqsubseteq, D_i \rangle$ as shown in Figure 2.

This defines, for each agent a, a function $h^a : I \cup \mathbb{I} \rightarrow \mathcal{D}$ which assigns a decision to each object. The decision d_i for object o is found using the correspondence $\langle C_i^a, \sqsubseteq, D_i \rangle$ attached to the most specific class C_i^a to which o belongs to ontology O^a.

4.4 Tasks and Interactions

Agents perform tasks in the environment by classifying objects according to their decisions. Following this, the agent receives a payoff that reflects the correctness of its decisions at the classification task. An agent a performs a task by going through the following steps:

(1) A subset S of objects with different properties from I is presented to agent a. The proportion t of S over $\mathbb{I}$ is called the task ratio.
(2) Agent a labels the objects $o \in S$ of the set with the decisions $h^a(o)$.
(3) Agent a receives the payoff $r_{S,a}$ reflecting how well it performed the task. We choose to take the number of objects labelled correctly, i.e. $r_{S,a} = |\{o \in S; h^a(o) = h^*(o)\}|$, as payoff.

Two agents a and b are able to interact with each other about an object o by going through the following steps:

(1) agents a and b disclose their decisions $h^a(o)$ and $h^b(o)$.
(2) if $h^a(o) = h^b(o)$ then they agree (success),
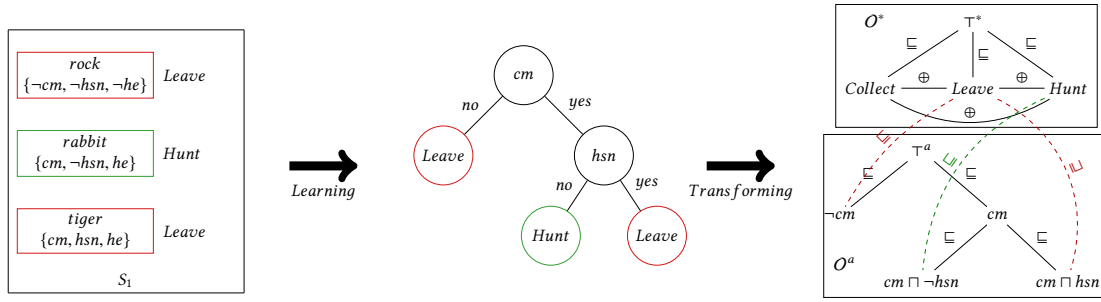(3) otherwise they do not agree (failure).

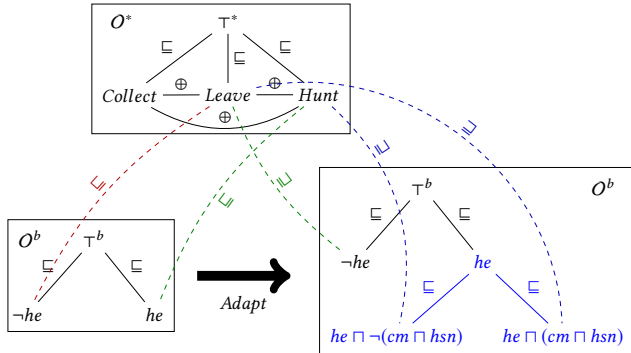Figure 2: Example of how an agent learns a decision tree-like ontology from a set of samples.

Figure 3: Example of agent b adaptation with $C_w \equiv (cm \sqcap hsn)$ by adding two classes: $he \sqcap (cm \sqcap hsn)$ and $he \sqcap \neg(cm \sqcap hsn)$. It also removes $\langle he, \sqsubseteq, Hunt \rangle$ and adds $\langle he \sqcap \neg(cm \sqcap hsn), \sqsubseteq, Hunt \rangle$ and $\langle he \sqcap (cm \sqcap hsn), \sqsubseteq, Leave \rangle$.

4.5 Adaptation

After a failure in communication, agents first compare their respective payoffs to determine which agent should adapt its ontology. Let agent w be the one having the higher payoff and agent l the one with the lower one (if they have an equal payoff, one of them will be selected randomly as agent w).

Then, agent l adapts its ontology. Let C_w (resp. C_l) be the leaf class to which object o belongs in O^w (resp. O^l). The effect of the adaptation operator is to split the objects of class C_l into two sets. The set of objects that belong to C_w will have their decision class changed to the one of C_w, and those which do not, will keep the same decision, as shown in Figure 3.

The adaptation happens as follows:

(1) Agent l asks agent w for the definition of its class C_w.
(2) If $C_l \sqcap \neg C_w \not\sqsubseteq \bot$ (or $C_l \not\sqsubseteq C_w$, i.e. some objects classified as C_l are not classified as C_w), agent l creates the classes $C_l^1 \equiv C_l \sqcap C_w$ and $C_l^2 \equiv C_l \sqcap \neg C_w$. Otherwise, it sets $C_l^1 = C_l$.
(3) Let D_l be the decision class for C_l and D_w be the decision class for C_w. Agent l replaces $\langle C_l, \sqsubseteq, D_l \rangle$ by $\langle C_l^1, \sqsubseteq, D_w \rangle$. If C_l^2 has been created, it also adds $\langle C_l^2, \sqsubseteq, D_l \rangle$.

5 EXPERIMENTS

To answer the raised questions, we run experiments based on the presented framework. The questions are reformulated as hypotheses tested through a systematic experiment plan whose output is measured.

5.1 Hypotheses

We test three hypotheses corresponding to the raised questions from Section 3.

- **Hypothesis 1:** The success rate converges to 1. This corresponds to the fact that, after a while, communication is always successful.
- **Hypothesis 2:** The average accuracy of the population improves at the end of the experiment.
- **Hypothesis 3:** Agents do not necessarily converge to the same ontologies, i.e. the average ontology distance of the experiments is not necessarily 0.

5.2 Experiment Plan

In one run of the experiment, agents initially learn from the environment how to take decisions. Then, they go through n iterations of the following:

(1) A pair of agents is selected randomly.
(2) The two agents perform a task in the environment for which they receive payoff.
(3) They interact to take a decision about one object selected randomly from the environment.
(4) Agents adapt their ontologies if the interaction failed.

Since the experiment depends on different factors, mentioned in Section 4, we define an experiment plan to vary these parameters as presented in Table 1 and run each combination 10 times. This means that we processed $q = 5 \times 3 \times 3 \times 3 \times 4 \times 10 = 5400$ simulations of 40000 iterations. At each iteration, the measures defined in Section 5.3 are recorded.

5.3 Measures

To assess the three presented hypotheses, we define: (1) the interaction success rate which indicates how often agents have agreed on their decision, (2) the accuracy of agents' classifiers on $\mathbb{I}$ and (3) a distance measure between ontologies. An experiment E^p is identified by its identifier p and characterised by the tuple of parameters $\langle A_p, \mathcal{P}_p, \mathcal{D}_p, r_p, t_p, n_p \rangle$ as defined in Table 1. The state $E_k^{p,j}$

Meaning	Variable	Range
Number of agents	$|A|$	$\{2, 5, 10, 20, 40\}$
Number of properties	$|\mathcal{P}|$	$\{3, 4, 5\}$
Number of decision classes	$|\mathcal{D}|$	$\{2, 3, 4\}$
Training ratio	r	$\{0.1, 0.3, 0.5\}$
Task ratio	t	$\{0.2, 0.4, 0.6, 0.8\}$
Number of iterations	n	40000

Table 1: Independent variable ranges

of agent k at iteration j in experiment p is described by its ontology $O_k^{p,j}$ at that iteration. We define the following measures computed at each iteration for each agent:

- $accuracy(E_k^{p,j})$ is the accuracy of agent k's ontology with respect to the set $\mathbb{I}$, i.e. the ratio of objects in $\mathbb{I}$ that are well classified by agent k's ontology to all objects of $\mathbb{I}$.

$$accuracy(E_k^{p,j}) = \frac{|\{o \in \mathbb{I}| h_k^{p,j}(o) = h^*(o)\}|}{|\mathbb{I}|}$$

such that $h_k^{p,j}(o)$ is the decision of agent k for object o in state $E^{p,j}$.

- the distance between two ontologies O_a and O_b is

$$\delta(O_a, O_b) = 1 - \frac{eq(O_a, O_b)}{max(|O_a|, |O_b|)}$$

$eq(O_a, O_b)$ is the number of equivalent classes between ontologies O_a and O_b, i.e. the number of classes that are defined equivalently in terms of property values:

$$eq(O_a, O_b) = |\{(C_a, C_b) \in O_a \times O_b | O_a, O_b \models C_a \equiv C_b\}|$$

For each experiment state $E^{p,j}$, we measure the success rate ($srate(E^{p,j})$) as the ratio of successful interactions until j, the accuracy ($accuracy(E^{p,j})$) as the average accuracy for all agents, and the distance ($distance(E^{p,j})$) as the average distance between each pair of distinct agent ontologies.

5.4 Comparison with an Alternative Approach

To determine how agents perform on realistic data compared to a coordinated learning approach, we repeated the experiment by generating the environment from an existing classification dataset. We used the Zoology dataset from the UCI machine learning repository [11] because its attributes are easily converted to binary. The obtained results were compared with those of A-MAIL [19].

The only differences with the settings of the main experiment are: (a) The environment objects and their decisions are taken from the dataset instead of generated randomly. We performed 10-fold cross-validation, thus using only 90% of the dataset as environment objects. (b) The task ratio is fixed to 0.2, the lowest value from the previous experiment. The training ratio is also set to 0.2 which corresponds to the ratio agents in A-MAIL use for training.

As mentioned in Section 2, A-MAIL is a coordinated inductive learning approach where agents engage in an explicit argumentation process on what they learned to reach a common classification. Agents do not have access to the same information in the two approaches. In A-MAIL agents keep the datasets they learned from

in memory and are able to use them in argumentation. However, in our setting, agents do not keep their datasets but receive an evaluation from the environment, i.e. the payoff corresponding to their performances on achieving tasks. This is why the comparison is merely indicative.

We measure, in addition to the accuracy, the precision and recall of experiment p at iteration j. The precision (resp. recall) of agent k with respect to decision d is the ratio of objects of decision d that agent k correctly classifies to the objects that agent k classifies in decision d (resp. to the objects for which decision d is correct):

$$precision(E_k^{p,j}, d) = \frac{|I_{k,d}^{p,j} \cap I_d|}{|I_{k,d}^{p,j}|} \qquad recall(E_k^{p,j}, d) = \frac{|I_{k,d}^{p,j} \cap I_d|}{|I_d|}$$

such that $I_{k,d}^{p,j} = \{o \in \mathbb{I}| h_k^{p,j}(o) = d\}$ and $I_d = \{o \in \mathbb{I}| h^*(o) = d\}$. The precision and recall are averaged per decision class and then per agents. As usual, the F-measure is the harmonic mean of these precision and recall.

6 RESULTS AND DISCUSSION

Table 2 contains the average of success rate, accuracy and ontology distance, grouped by factor values, at the first, intermediate and final iterations for all the experiments. The table includes the measures after the first interaction because some factors influence them from the start. For example, a larger training ratio corresponds to a higher success rate because agents have higher chances of getting overlapping training sets with larger training ratios and thus agree more on their decisions. It also includes some of the intermediate results which show that for some parameters, agents converge much faster to a stable state with successful communication. For instance, it can be observed that the success rate converges to 1 faster with fewer agents.

In what follows, we show how these results answer the three hypotheses. We also analyze the effects of different factors on the obtained results and discuss the main ones. Finally, the results of the experiment presented in Section 5.4 are provided.

6.1 Success Rate

Figure 4 shows the average success rate at each iteration.
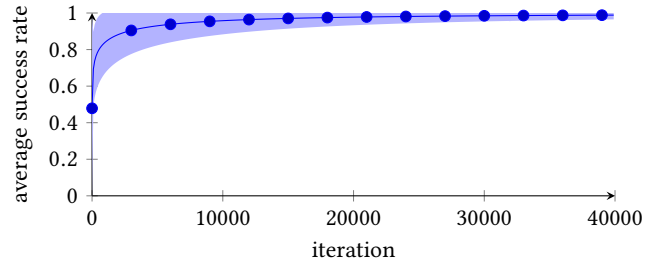
Figure 4: Average success rate over 40000 iterations. The shaded part boundaries represent the standard deviation from the average.

The success rate converges to 1, which supports the first hypothesis. The standard deviation gradually decreases as the number of

|		number of agents					number of properties			number of classes			training ratio			task ratio			
		2	5	10	20	40	3	4	5	2	3	4	0.1	0.3	0.5	0.2	0.4	0.6	0.8
srate	1	0.47	0.48	0.51	0.46	0.47	0.50	0.47	0.47	0.58	0.48	0.37	0.43	0.47	0.54	0.47	0.47	0.48	0.50
	2000	**1.00**	0.98	0.91	0.81	0.71	0.94	0.89	0.81	0.92	0.88	0.85	0.90	0.86	0.88	0.86	0.88	0.89	0.90
	10000	**1.00**	**1.00**	0.98	0.94	0.87	0.99	0.97	0.92	0.97	0.96	0.94	0.96	0.95	0.96	0.94	0.96	0.96	0.97
	40000	**1.00**	**1.00**	**1.00**	0.99	0.96	**1.00**	0.99	0.98	**0.99**	**0.99**	0.98	**0.99**	**0.99**	**0.99**	0.98	**0.99**	**0.99**	**0.99**
accuracy	1	0.57	0.56	0.56	0.56	0.56	0.58	0.56	0.56	0.66	0.54	0.49	0.45	0.56	0.68	0.56	0.56	0.56	0.57
	2000	0.61	0.70	0.79	0.82	0.79	0.78	0.75	0.70	0.81	0.73	0.69	0.59	0.77	0.86	0.70	0.74	0.75	0.78
	10000	0.61	0.70	0.80	0.88	0.92	**0.79**	0.78	0.77	**0.84**	0.77	0.74	0.63	0.82	**0.90**	0.75	0.78	0.79	**0.81**
	40000	0.61	0.70	0.80	0.88	**0.94**	**0.79**	0.78	**0.79**	**0.84**	0.77	0.74	0.64	0.82	**0.90**	0.76	0.78	0.79	**0.81**
distance	1	0.56	0.61	**0.62**	**0.62**	0.61	0.43	0.61	**0.77**	0.58	0.61	**0.62**	0.39	0.69	**0.73**	**0.61**	0.60	0.60	0.60
	2000	0.47	0.47	0.47	0.49	0.57	0.33	0.50	0.65	0.52	0.49	0.47	0.38	0.55	0.55	0.49	0.49	0.50	0.50
	10000	0.47	0.47	0.47	0.47	0.48	0.33	0.49	0.60	0.50	0.47	0.44	0.35	0.52	0.54	0.46	0.47	0.48	0.48
	40000	0.47	0.47	0.47	0.47	0.48	0.33	0.49	0.60	0.50	0.47	0.44	0.35	0.52	0.54	0.46	0.47	0.48	0.48

Table 2: Average of success rate, accuracy and ontology distance at the first (1), last (40000) and intermediate (2000, 10000) iterations of all experiments grouped by factor values. In bold, the highest values of the cell.

iterations increases. This indicates that the success rate of different simulation runs are converging similarly even though they start at different levels due to randomness in initial ontologies or different simulation factors.

6.2 Accuracy

Figure 5 shows the difference in distributions of average agent accuracies at the start of the simulations and at their end. The distribution at the end of the simulations shifts towards 1 which indicates an overall improvement of agent accuracies. Table 2 shows that this happens for all factor values. This corroborates a weak version of the second hypothesis, i.e. on average on all the runs accuracy improves.

To show that the difference in average accuracy is significant, we conducted a paired Student t-test between the average accuracy at the beginning of the simulation (Mean= 0.56, Standard Deviation= 0.14) and at the end (Mean= 0.79, Standard Deviation= 0.2). There is a significant difference with $t = 100.06$ and $p < 0.01$.
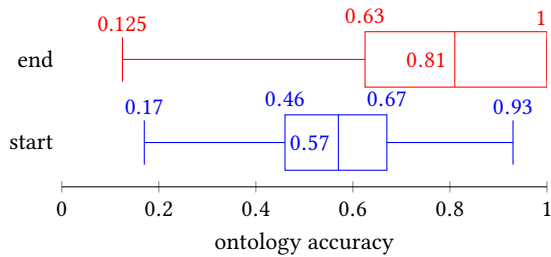
Figure 5: Distribution of accuracies at the start of the simulation (blue) and at the end of the simulation (red).

However, the left-hand extremity of Figure 5 tells us that there are cases in which accuracy actually decreases. This rebuts a strong version of the hypothesis, i.e. that accuracy increases at each run. In 3.5% of the runs the final average accuracy is lower than the initial one. This is explained by agents having a rare correct decision

which is lost because they disagree with others which have gathered more payoff. Consider two agents a and b that classify all objects correctly except that a classifies o incorrectly and b classifies o' incorrectly. It may happen that agent a performs better than agent b in the task because S contains o' but not o; a would then gather more payoff than b. If o is the selected object, then as a result b will change its (correct) decision about o to the (incorrect) one held by a. If these are the only two agents, they now have ontologies whose average accuracy is lower. Section 6.4.3 discusses this further.

6.3 Ontology Distance

The boxplots in Figure 6 show the distribution of average ontology distances at the start of the simulation and at the end of it. We observe that the distribution slightly shifts towards 0 at the end of the simulation. As expected, agents end up with more similar ontologies. However, they do not necessarily share the same ontologies. Table 2 shows that for all factor values, the average distance remains far from 0. In fact, 90.78% of the runs do not lead to the same ontologies. The reason behind this is that agents may consider different properties to take the same decision. Figure 7 shows the ontologies of agents a and b who take the same decisions for all objects. For example, both agents a and b would take decision D_2 for object o that has the properties $\{\neg p_1, p_2, p_3\}$. Agent a would take it because o has p_2 and agent b because it has $\neg p_1$. This supports the third hypothesis.

6.4 Effects of Simulation Factors

To determine which factors (independent variables) significantly affect which measures (dependent variables) we performed an analysis of variance (ANOVA) test on the final measured values of success rate, distance and accuracy of each simulation run. The following results are simplified as we study only the direct effect of each factor on its own (contrary to N-way ANOVA). ANOVA returns, for each pair of independent and dependent variables, the probability that the independent variable has no effect on the dependent variable (p-value). We consider a p-value that is lower than 0.01 as low enough to reject that the independent variable has no
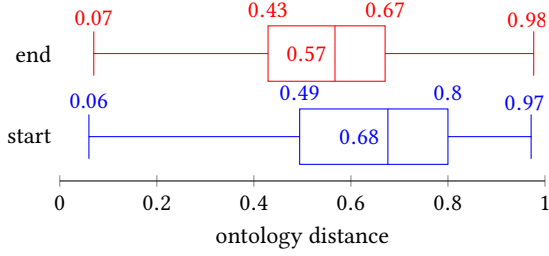
Figure 6: Distribution of average ontology distances at the start of the simulation (blue) and at the end of the simulation (red).
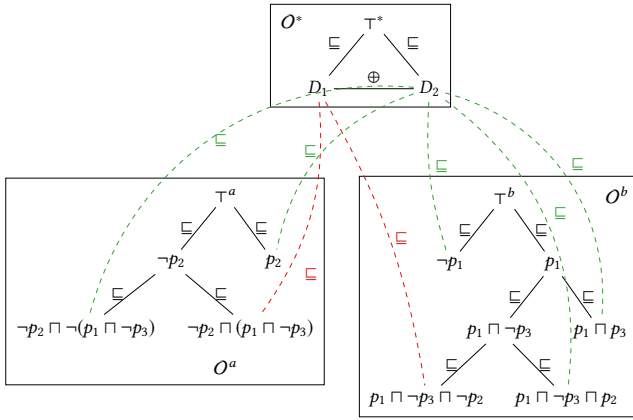

Figure 7: Example of two ontologies O^a and O^b that take the same decisions with only two equivalent classes.

effect on the dependent variable. ANOVA resulted in a significant effect of all factors on all measures except for (a) the number of agents on the distance and (b) the number of properties on the accuracy. However, ANOVA only informs if there is an effect from the independent variables. To know how an independent variable affects a dependent variable we performed a post-hoc Tukey HSD (honestly significant difference) test. In the following, we discuss three of the main effects. The full analysis can be found in our experiment logbook [8].

6.4.1 Effect of number of agents on accuracy. On the one hand, Figure 8 shows that the number of agents does not significantly affect the final ontology distance. On the other hand, it shows that the more agents there are, the higher the final accuracy is. This is because correct pieces of knowledge could be completely lost if they are part of an overall bad agent knowledge. To illustrate this, consider an object o of decision d_j such that it is classified correctly by agents with lower payoffs. If the object is picked up in the early iterations, the agents may wrongly correct it, and since there are only few agents, the information "o has the decision d_j" might completely disappear, as discussed in Section 6.2. However, with more agents, the chances that this information survives for later iterations are higher. Thus, agents with low payoffs get the
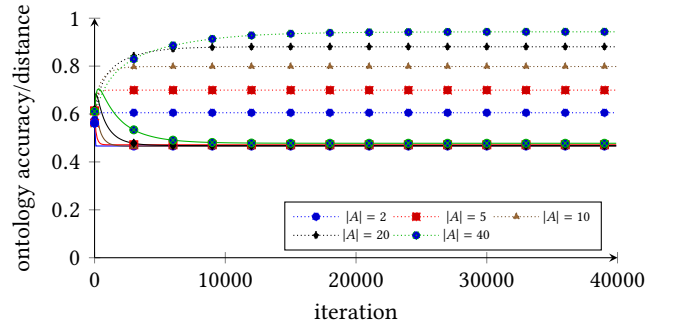

Figure 8: Average ontology accuracy (dotted) and ontology distance (plain) by number of agents $|A|$.

chance to correct their other errors to increase their payoffs and start spreading the information that o is of decision d_j.

6.4.2 Effect of number of properties on ontology distance. On the one hand, Figure 9 shows that the final average accuracy is not significantly affected by the number of properties. On the other hand, the average ontology distances at the beginning of the simulation are at different values, higher for the higher number of properties, and they all drop by about the same amount. As the number of properties increases, agents have more flexibility on which properties they consider to take decisions, which results in them having different class definitions and still agree on the decisions. For example, if all objects having the property p_1 are classified as d_1 and the rest as d_2, agent a might consider directly the property p_1 to take the decision while agent b can first consider the property p_2 then the property p_1. As a result, the agents would have different class definitions. If one of the properties is removed, both agents will have to use the same property which would result in them having equivalent class definitions.
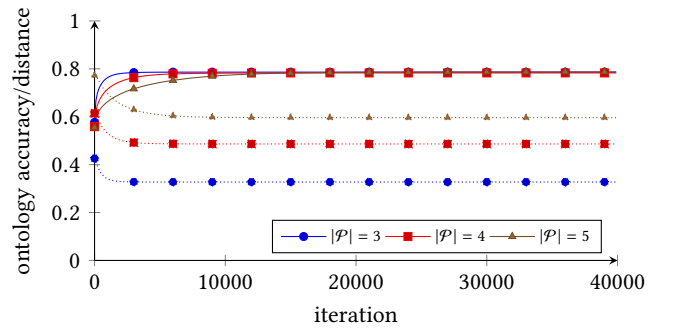

Figure 9: Average ontology accuracy (plain) and ontology distance (dotted) by number of properties $|\mathcal{P}|$.

6.4.3 Effect of task ratio and number of agents on accuracy difference. Both a higher task ratio and a larger number of agents lead to higher accuracy on average (Table 2). However, as noted in Section 6.2, the accuracy difference between the final and initial iteration is not always positive. We identified the task ratio and the number of

agents as the main factors that influence this. As can be observed from Table 3, a lower task ratio and fewer agents increase the chances of this to happen.

$t \setminus |A|$	2	5	10	20	40	total
0.2	53	29	2	0	0	84
0.4	43	9	0	0	0	52
0.6	32	6	2	0	0	40
0.8	13	0	0	0	0	13
total	141	44	4	0	0	189

Table 3: Number of runs with negative accuracy difference by number of agents and task ratio (each cell = 360 runs).

With a higher task ratio, the received payoff better assesses the quality of the decisions taken by the agents. Thus agents have less chances of changing their decisions to a wrong one. However, even with a low task ratio, if the agent population is large, lost decisions are more easily recovered. This is clearly illustrated by Table 3. Obviously, if the problem illustrated in Section 6.2 occurs with 2 agents, the correct decision is definitively lost. When there are more agents, it may be recovered.

6.5 Results on Real Data

Table 4 displays the results obtained in the additional experiment of Section 5.4.

Method	$|A|$	Precision	F-measure	Recall	Accuracy
Simulation	2	0.88	0.87	0.86	0.951
	5	0.91	0.89	0.88	0.964
	10	0.94	0.92	0.91	0.977
	20	0.96	**0.94**	**0.93**	**0.984**
	40	0.95	**0.94**	**0.93**	0.983
A-MAIL	2	0.97	0.85	0.75	0.950
	3	**0.98**	0.89	0.81	0.968
	4	0.97	0.90	0.84	0.966
	5	**0.98**	0.93	0.88	0.980

Table 4: Final precision, F-measure, recall and accuracy of different methods on the test set.

When A-MAIL is used with 2, 3 and 4 agents, they only learn from 40%, 60% and 80% of the training dataset respectively, which explains the relatively low results compared to 5 agents in which 100% of the training set is used. A-MAIL results do not improve with more agents [19]. In contrast, the performance of agents presented here depends more on their number. With enough agents, they can improve their knowledge significantly to reach results on par with A-MAIL.

Agents perform better in a realistic dataset than on randomly generated objects and decisions. With 2 agents, the accuracy on a realistic dataset (16 binary properties and 7 decision classes) is 0.95 compared to an average of 0.61 in randomly generated datasets (3 to 5 binary properties and 2 to 4 decision classes). A similar improvement can be observed for the other numbers of agents. This is due to feature patterns existing in the real classes and not

in randomly generated ones making the generalisation easier for agents.

7 CONCLUSION

A first contribution of this paper is the experimental framework in which agents first learn the knowledge that they later adapt. This shows that it is possible to have continuity between these two tasks. This also provides an experimental setting in which it is possible to independently experiment with different ontology learning and adaptation methods.

Nonetheless, the main contribution is the experimental test of the three hypotheses: (a) agents can adapt knowledge, improving agreement and communication, (b) doing so they, most of the time, develop more accurate knowledge, and (c) this does not constrain them to have the same knowledge. Under specific conditions and in 3.5% of the cases, knowledge may be forgotten. This is realistic and it would be worth considering how agents could address this.

Finally, we tested this approach on a classification task with realistic data and contrasted the obtained results with a coordinated inductive learning approach. The achieved correctness and completeness of both approaches were on par.

This work shows how agents can globally evolve their knowledge by locally reacting to environment and society pressure. It constitutes a step towards agents able to effectively adapt to other individuals and their environment.

Several things were left aside in this paper. The payoff structure used to assess agents is uniform. However, in reality this is not always the case, e.g. the payoff received by running from a lion should be higher than the one received from collecting an apple. Such payoff structure may further constrain the evolution of agent ontologies. In addition, agents had perfect shared knowledge about the object properties and the possible decisions. In reality, there might be differences on how agents perceive the environment and what decisions they take. It would be worth exploring these issues.

This work is not exclusive of other means such as developing alignments and adapting them. Future work may explore the impact of different learning methods as well as different agent behaviours, i.e. other adaptation operators or reinforcement learning, combined with other assessment methods that can originate from the environment, the society or both.

DATA AVAILABILITY

All experiments were performed in the *Lazy lavender* software environment [1]. Settings, output and data analysis notebooks are made available at [8] and [7].

REFERENCES

[1] 2020. *Lazy lavender*. https://gitlab.inria.fr/moex/lazylav.
[2] Kemo Adrian and Enric Plaza. 2017. An approach to interaction-based concept convergence in multi-agent systems. In *Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology, Bozen-Bolzano, Italy, September 21-23, 2017 (CEUR Workshop Proceedings, Vol. 2050)*. CEUR-WS.org.

[3] Kemo Adrian and Enric Plaza. 2019. Argumentation on meaning: a semiotic model for contrast set alignment. In *Proceedings of the Joint Ontology Workshops 2019 Episode V: The Styrian Autumn of Ontology, Graz, Austria, September 23-25, 2019 (CEUR Workshop Proceedings, Vol. 2518)*. CEUR-WS.org.

[4] Michael Anslow and Michael Rovatsos. 2015. Aligning experientially grounded ontologies using language games. In *Graph Structures for Knowledge Representation and Reasoning - 4th International Workshop, GKR 2015, Buenos Aires, Argentina, July 25, 2015, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 9501)*. Springer, 15–31.

[5] Manuel Atencia and Marco Schorlemmer. 2012. An interaction-based approach to semantic alignment. *Journal of Web Semantics* 12 (2012), 131–147.

[6] Franz Baader, Diego Calvanese, Deborah Mcguinness, Daniele Nardi, and Peter Patel-Schneider. 2007. *The description logic handbook: Theory, implementation and applications* (2 ed.). Cambridge University Press.

[7] Yasser Bourahla. 2020. 20201001-DOLA Experiment description. https://doi.org/10.5281/zenodo.4507093 http://sake.re/20201001-DOLA.

[8] Yasser Bourahla. 2020. 2020623-DOLA Experiment description. https://doi.org/10.5281/zenodo.4507546 http://sake.re/20200623-DOLA.

[9] Leonid L. Chepelev, Dana Klassen, and Michel Dumontier. 2011. Chemical hazard estimation and method comparison with OWL-Encoded toxicity decision trees. In *Proceedings of the 8th International Workshop on OWL: Experiences and Directions (OWLED 2011), San Francisco, California, USA, June 5-6, 2011 (CEUR Workshop Proceedings, Vol. 796)*.

[10] Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems (Lecture Notes in Computer Science, Vol. 1857)*. Springer Berlin Heidelberg, 1–15.

[11] Dheeru Dua and Casey Graff. 2017. UCI machine learning repository. http://archive.ics.uci.edu/ml

[12] Jérôme Euzenat. 2017. Interaction-based ontology alignment repair with expansion and relaxation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. ijcai.org, 185–191.

[13] Jérôme Euzenat and Pavel Shvaiko. 2013. *Ontology matching* (2nd ed.). Springer Publishing Company, Incorporated.

[14] N. Jennings. 1993. Commitments and conventions: The foundation of coordination in multi-agent systems. *Knowl. Eng. Rev.* 8 (1993), 223–250.

[15] N. Jennings, P. Faratin, A. Lomuscio, S. Parsons, M. Wooldridge, and C. Sierra. 2000. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation* 10 (2000), 199–215.

[16] Eric H Y Lau, Benjamin J Cowling, Lai-Ming Ho, and Gabriel M Leung. 2008. Optimizing use of multistream influenza sentinel surveillance data. *Emerging Infectious Diseases* 14, 7 (2008), 1154–1157.

[17] Sergio Manzano, Santiago Ontañón, and Enric Plaza. 2012. A case-based approach to mutual adaptation of taxonomic ontologies. In *Case-Based Reasoning Research and Development - 20th International Conference, ICCBR 2012, Lyon, France, September 3-6, 2012. Proceedings (Lecture Notes in Computer Science, Vol. 7466)*. Springer, 226–240.

[18] Zaruhi R. Mnatsakanyan, Howard S. Burkom, Jacqueline S. Coberly, and Joseph S. Lombardo. 2009. Bayesian information fusion networks for biosurveillance applications. *Journal of the American Medical Informatics Association : JAMIA* 16, 6 (Nov. 2009), 855–863.

[19] Santiago Ontañón and Enric Plaza. 2015. Coordinated inductive learning using argumentation-based communication. *Autonomous Agents and Multi-Agent Systems* 29, 2 (2015), 266–304.

[20] Terry R. Payne and Valentina A. M. Tamma. 2014. Negotiating over ontological correspondences with asymmetric and incomplete knowledge. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS'14, Paris, France, May 5-9, 2014*. IFAAMAS/ACM, 517–524.

[21] John Ross Quinlan. 1986. Induction of decision trees. *Mach. Learn.* 1, 1 (March 1986), 81–106.

[22] Stuart Russell and Peter Norvig. 2009. *Artificial intelligence: A modern approach* (3rd ed.). Prentice Hall Press, USA.

[23] Dymitr Ruta and Bogdan Gabrys. 2000. An overview of classifier fusion methods. *Computing and Information Systems* 7 (2000), 1–10.

[24] Luc Steels. 2012. *Experiments in cultural language evolution*. John Benjamins Publishing Company.

[25] Gaëtan Texier, Rodrigue S. Allodji, Loty Diop, Jean-Baptiste Meynard, Liliane Pellegrin, and Hervé Chaudet. 2019. Using decision fusion methods to improve outbreak detection in disease surveillance. *BMC Medical Informatics and Decision Making* 19, 1 (2019), 38–49.

[26] Cássia Trojahn, Jérôme Euzenat, Valentina Tamma, and Terry R. Payne. 2011. *Argumentation for reconciling agent ontologies*. Springer Berlin Heidelberg, Berlin, Heidelberg, 89–111. https://doi.org/10.1007/978-3-642-18308-9_5

[27] Line van den Berg, Manuel Atencia, and Jérôme Euzenat. 2020. Agent ontology alignment repair through dynamic epistemic logic. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*. International Foundation for Autonomous Agents and Multiagent Systems, 1422–1430.

[28] Jurriaan van Diggelen, Robbert-Jan Beun, Frank Dignum, Rogier M. van Eijk, and John-Jules Ch. Meyer. 2006. ANEMONE: An effective minimal ontology negotiation environment. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*. ACM, 899–906.

[29] Nan Zhi, Terry R. Payne, Piotr Krysta, and Minming Li. 2019. Truthful mechanisms for multi agent self-interested correspondence selection. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11778)*. Springer, 733–750.