Vocabulary Alignment in Openly Specified Interactions

Paula Chocron
IIIA-CSIC
Universitat Autònoma de Barcelona
Bellaterra, Catalonia, Spain
pchocron@iiia.csic.es

Marco Schorlemmer
IIIA-CSIC
Bellaterra, Catalonia, Spain
marco@iiia.csic.es

ABSTRACT

The problem of achieving common understanding between agents that use different vocabularies has been mainly addressed by designing techniques that explicitly negotiate mappings between their vocabularies, requiring agents to share a meta-language. In this paper we consider the case of agents that use different vocabularies and have no meta-language in common, but share the knowledge of how to perform a task, given by the specification of an interaction protocol. For this situation, we present a framework that lets agents learn a vocabulary alignment from the experience of interacting. Unlike previous work in this direction, we use open protocols that constrain possible actions instead of defining procedures, making our approach more general. We present two techniques that can be used either to learn an alignment from scratch or to repair an existent one, and we evaluate experimentally their performance.

Keywords

Agent Communication, Vocabulary Alignment, Open Interaction Protocols

1. INTRODUCTION

Addressing the problem of vocabulary heterogeneity is necessary for the common understanding of agents that use different languages, and therefore crucial for the success of multi-agent systems that act jointly by communicating. This problem has been tackled several times in the past two decades, in general from two different perspectives. Some approaches [21, 8] consider the existence of external *contextual* elements that all agents perceive in common, and explore how those can be used to explain the meaning of words. A second group of techniques [16, 19] (and also [8], where both approaches are combined) consider the situation, reasonable in agents that communicate remotely, in which this kind of context is not available. They do so by providing explicit ways of learning or agreeing on a common vocabulary (or alignment between vocabularies). These techniques require agents to share a common meta-language that they can use to discuss about the meaning of words and their alignments. The complex question of how to communicate with
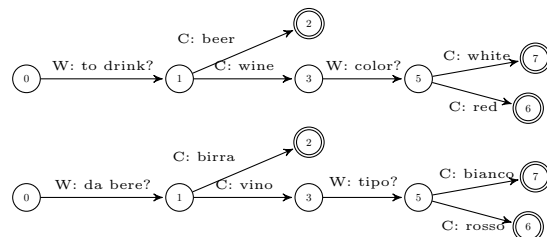
Figure 1: Protocols for Waiter (W) and Customer (C)

heterogeneous interlocutors when neither a physical context nor a meta-language are available remains practically unexplored.

The work by Atencia and Schorlemmer [1] approaches this situation by considering a different kind of context, given by the interactions in which agents engage. Agents are assumed to share the *knowledge of how to perform a task*, or, more concretely, the specification of an interaction, given by a finite state automaton. For example, they consider the interaction in Figure 1, in which an English speaking customer and an Italian waiter communicate to order drinks. The authors show how agents can progressively learn which mappings lead to successful interactions from the experience of performing the task. After several interactions, agents converge to an alignment that they can use to always succeed at ordering and delivering drinks with that particular interlocutor. The interesting aspect of this idea is that the only shared element, namely the interaction specification, is already necessary to communicate. However, using finite state automata as specifications implies that agents need to agree on the exact order in which messages should be sent, which is unnecessarily restrictive. In addition, agents learn an alignment that is only useful for one task, and how it can be extrapolated to further interactions is not explored.

In this paper we present a general version of the *interaction as context* approach to vocabulary alignment in multi-agent systems. Instead of automata, we consider agents that specify interactions with constrained-based *open protocols* that define rules about what can be said instead of forcing one particular execution. These protocols provide a much more flexible specification, that does not define the complete flow of the messages but only some relevant rules. In particular, we use ConDec protocols [15] that use linear temporal logic constraints. One ConDec protocol may not define completely the meaning of all words in a vocabulary,

so we study how agents can learn mappings from performing different tasks. For this reason, the learning process is substantially different to the one in [1], where a mapping that works in one task is considered correct. If enough experiences are available, agents that use the techniques we present converge to an alignment that is useful in the general case, even for future interactions that they do not yet know. Since learning is done gradually as different interaction opportunities appear, agents can use what they learned early, even if they still do not know the alignment completely.

After presenting *open interaction protocols*, we define a framework for interacting with partners that use different vocabularies. We later present two different techniques to learn an alignment from the experience of interacting. The first relies only on analysing if a message is allowed or not in a particular moment, and it can be used in any constraint-based protocol. The second technique uses the semantics of the protocols we present to improve the performance of the first one. Both methods can be used to learn alignments from scratch when there is no information, as well as to repair alignments obtained with other methods. We evaluate experimentally the techniques when used for both purposes, and show how different factors affect their performance.

2. OPEN INTERACTION PROTOCOLS

The question of what should be expressed by an interaction protocol has been extensively discussed in the multi-agent systems community. Traditional approaches such as finite state automata and Petri Nets are simple to read and verify, but also too rigid. More flexible alternatives constrain possible actions instead of defining a fixed procedure. *Constraint Declarative Protocols* (commonly known as ConDec protocols) are an example of these approaches. ConDec protocols were first proposed by Pesic and van der Aalst [15] as a language to describe business protocols, and then used as specifications for agent interactions by Montali [14], who presented an extension of ConDec and tools for its verification, and by Baldoni et al. [2, 3], who integrated ConDec constrains and commitment protocols, a framework to specify interactions with social semantics first proposed by Singh [20]. An important advantage of ConDec protocols is that they use linear temporal logic, a well-known formalization for which many tools are available. On the negative side, it has been argued that they specify only the temporal aspect of the interaction, without any link to a more global view of the task that is being performed [13]. However, we think these temporal constraints are a good starting point to study the use of interaction specifications for vocabulary alignment.

Linear temporal logic (LTL from now on) is a natural choice to express constraints about actions that occur in time. The syntax of LTL includes the one of propositional logic, and the additional temporal operators $\{\Box, \Diamond, \bigcirc, U\}$, that can be applied to any LTL sentence. LTL sentences are interpreted over *paths* in Kripke structures, which are sequences of states associated to a truth-valuation of propositional variables. Temporal operators are interpreted in these paths as follows: $\Box p$ means that p must be true in the truth-valuation of all states, $\Diamond p$ means that p must be true eventually, $\bigcirc p$ means that p must be true in the next state, and $p\,U\,q$ means that p must be true until q is true. A set of LTL sentences, called a *theory*, is *satisfiable* if there exists a path for which all the sentences are true. In that

case, the path is a *model* of the theory. The satisfiability problem in LTL is decidable, as well as the *model checking* problem, which consists in deciding if a given path is a model of a theory.

A ConDec protocol is a tuple $\langle M, C \rangle$, where M is a set of action names and C is a set of constraints on how actions can be performed. Constraints are LTL sentences renamed conveniently. We use a minimal version[1] of the constraints introduced originally by Pesic and van der Aalst that we show in Table 1, where $n \in \mathbb{N}$ and $a, b \in M$. These constraints are generally divided into three classes. Existential constraints (*existence* and *absence*) predicate over the amount of times some action can be performed, while relational constraints describe binary relations between two, either positive or negative (starting with !). In a protocol $\langle M, C \rangle$, $Cons(M)$ is the set of all possible constraints over M, and $C \subseteq Cons(M)$.

Constraint	LTL meaning
$existence(1, a)$	$\Diamond a$
$existence(n + 1, a)$	$\Diamond(a \wedge \bigcirc existence(n, a))$
$absence(n, a)$	$\neg existence(n + 1, a)$
$correlation(a, b)$	$\Diamond a \rightarrow \Diamond b$
$!correlation(a, b)$	$\Diamond a \rightarrow \neg \Diamond b$
$response(a, b)$	$\Box(a \rightarrow \Diamond b)$
$!response(a, b)$	$\Box(a \rightarrow \neg \Diamond b)$
$before(a, b)$	$\neg b \, U \, a$
$!before(a, b)$	$\Box(\Diamond b \rightarrow \neg a)$
$premise(a, b)$	$\Box(\bigcirc b \rightarrow a)$
$!premise(a, b)$	$\Box(\bigcirc b \rightarrow \neg a)$
$imm\_after(a, b)$	$\Box(a \rightarrow \bigcirc b)$
$!imm\_after(a, b)$	$\Box(a \rightarrow \bigcirc \neg b)$

Table 1: LTL definitions of constraints

2.1 Open Protocols as Interaction Protocols

In the rest of this section we present the technical notions that are necessary to use ConDec protocols as specifications of interactions between agents that may use different vocabularies. We first define *interaction protocols* which constrain the way in which agents can utter messages. We introduce the notion of *bound* to capture the knowledge of agents about how many steps they can use for the interaction. If this knowledge is not available it can be replaced for the fact that the interaction must finish in a finite number of steps.

DEFINITION 1. *Given a set A of agent IDs and a propositional vocabulary V, an* interaction protocol *is a tuple $\mathfrak{P} = \langle M, C, b \rangle$, where $M = A \times V$ is a set of messages formed by the ID of the sender agent and the term it utters, $C \subseteq Cons(M)$, and $b \in \mathbb{N}$ is the bound.*

The semantics of an interaction protocol is defined over interactions that represent a sequence of uttered messages.

---

[1]Since we are not interested in the usability of the protocols here, we do not include some constraints that work as syntactic sugar, such as $exactly(n, a)$, that can be replaced by including $existence(n, a)$ and $absence(n, a)$.

DEFINITION 2. *Given a set of messages M, an* interaction *$i \in M^*$ is a finite sequence of messages $m \in M$. The length of an interaction ($len(i)$) and the append operation ($i.m$) are defined in the same way as for sequences.*

An interaction i can be encoded into a Kripke structure path by including one propositional variable for each message $m \in M$ and considering the following sequence of states with truth-valuations. In states with index $0 \leq j < len(i)$, let the propositional variable for the j-th message in i be true, and all other propositional variables be false. For states after $len(i)$, let all propositional variables be false. With this construction, we can define the semantics of interaction protocols.

DEFINITION 3. *An interaction i is a* model *of an interaction protocol $\mathfrak{P} = \langle M, C, b \rangle$ (noted $i \models \mathfrak{P}$) if $len(i) \leq b$, and the Kripke path that encodes i is a model of C. An interaction i' is a* partial model *(noted $i' \models_p \mathfrak{P}$) of $\mathfrak{P}$ if it is a prefix of a model of $\mathfrak{P}$.*

Definition 3 implies that checking satisfiability of an interaction protocol is equivalent to checking LTL satisfiability.

As already mentioned, we are interested in agents that use different vocabularies, but share the *knowledge of how to perform a task*. In the rest of this section we define more precisely what that means. From now on, V_1 and V_2 are two possibly different vocabularies. Let us start by defining the notion of vocabulary alignment.

DEFINITION 4. *An* alignment *is a function between vocabularies $\alpha : V_2 \to V_1$. Given a set of agents A, α can be extended homomorphically to a function between:*

- *messages $M_1 = A \times V_1$ and $M_2 = A \times V_2$ ($\alpha : M_2 \to M_1$)*

- *constraints ($\alpha : Cons(M_2) \to Cons(M_1)$) and sets of constraints ($\alpha : 2^{Cons(M_2)} \to 2^{Cons(M_1)}$)*

- *interactions ($\alpha : M_2^* \to M_1^*$) and sets of interactions ($\alpha : 2^{M_2^*} \to 2^{M_1^*}$)*

In this paper, for simplicity, we will restrict to using only bijective alignments over vocabularies of the same size.

To capture the idea of *sharing the knowledge of how to perform a task*, we define the notion of *compatibility* between protocols, which consists simply on having the same models modulo an alignment. Given a protocol $\mathfrak{P} = \langle M, C, b \rangle$, we define the set of its models as $Int(\mathfrak{P}) = \{i \in M^* \text{ such that } i \models \mathfrak{P}\}$.

DEFINITION 5. *Two interaction protocols $\mathfrak{P}_1 = \langle M_1, C_1, b \rangle$ and $\mathfrak{P}_2 = \langle M_2, C_2, b \rangle$ with sets of messages $M_1 = A \times V_1$ and $M_2 = A \times V_2$ are* compatible *if there exists an alignment $\alpha : V_2 \to V_1$ such that*

$$Int(\mathfrak{P}_1) = \alpha(Int(\mathfrak{P}_2))$$

If we know the alignment α for which the condition holds, we can say they are compatible under α.

An Example.

Let us extend the example of Figure 1 by considering again a waiter W and a customer C and the *ordering drinks* situation. The vocabulary of the customer is $V_C = \{$to drink, beer, wine, water, size, pint, half pint$\}$, while the *Waiter* uses $V_W = \{$da bere, birra, vino, acqua, tipo, media, piccola$\}$. Consider the bijective alignment $\alpha : V_W \to V_C$ obtained by mapping each word in V_W with the word in the same position in V_C. We consider the following protocols to specify the *ordering drinks* interactions:

$$\mathfrak{P}_C = \langle \{W, C\} \times V_C,$$
$\{existence(\langle W, \text{to drink}\rangle, 1, 1), premise(\langle W, \text{to drink}\rangle, \langle C,$
$\text{beer}\rangle), premise(\langle W, \text{to drink}\rangle, \langle C, \text{water}\rangle),$
$premise(\langle W, \text{to drink}, \langle C, \text{wine}\rangle), response(\langle C, \text{beer}\rangle, \langle W, \text{size}\rangle),$
$premise(\langle W, \text{size}\rangle, \langle C, \text{halfpint}\rangle), premise(\langle W, \text{size}\rangle, \langle C, \text{pint}\rangle)\}, 5 \rangle$

$$\mathfrak{P}_W = \langle \{W, C\} \times V_W,$$
$\{existence(\langle W, \text{da bere}\rangle, 1, 1), premise(\langle W, \text{da bere}\rangle, \langle C,$
$\text{birra}\rangle), premise(\langle W, \text{da bere}\rangle, \langle C, \text{vino}\rangle),$
$premise(\langle W, \text{da bere}\rangle, \langle C, \text{acqua}\rangle), !correlation(\langle C, \text{birra}\rangle,$
$\langle W, \text{vino}\rangle), response(\langle C, \text{birra}\rangle, \langle W, \text{tipo}\rangle), premise(\langle W, \text{tipo},$
$\langle C, \text{piccola}\rangle, \rangle), premise(\text{tipo}), \langle C, \text{media}\rangle\}, 5 \rangle$

The two protocols above are not compatible under any bijective alignment, since the for the one of the customer there in a model in which the customer orders wine *and* beer, while the other protocol has only models in which the waiter only accepts one alcoholic beverage. If $!correlation(\text{beer, wine})$ is added to $\mathfrak{P}_C$, the resulting protocols would be compatible, in particular under α.

3. COMMUNICATING WITH HETEROGENEOUS PARTNERS

We focus on interactions between two agents a_1 and a_2 with vocabularies V_1 and V_2 respectively. During an interaction, an agent can send messages composed of words in its vocabulary and receive others from its interlocutor, or finish the communication if certain conditions hold. We assume messages are never lost and always arrive in order, and more strongly, that each message is received before the following one is uttered. This requires that agents agree in who speaks at each time, although we do not force them to follow any particular turn-taking pattern.

Agents interact to perform some task together, that each of them specifies with an interaction protocol. We assume a_1 and a_2 agree on a set of tasks that they can perform and on *how* they are performed, this is, their respective protocols for each task are compatible. Our agents use their vocabulary consistently throughout different tasks, i.e., the protocols for all tasks are compatible under the same alignment. From now on, we will use sets of messages $M_1 = \{a_1, a_2\} \times V_1$ and $M_2 = \{a_1, a_2\} \times V_2$. Let $\alpha : V_2 \to V_1$ be a bijective alignment such that, for each task that a_1 and a_2 can perform, they have protocols $\mathfrak{P}_1 : \langle M_1, C_1, b \rangle$ and $\mathfrak{P}_2 : \langle M_2, C_2, b \rangle$ respectively, and $\mathfrak{P}_1, \mathfrak{P}_2$ are compatible under α. We assume that whenever agents interact, they use protocols that correspond to the same task.

We present a general approach to learn the alignment α from the experience of interacting to perform different tasks sequentially. The methods we present are used by one agent individually, and do not require that its interlocutor uses

them as well. For simpliciry, from now on we adopt the perspective of agent a_1.

Briefly, we propose to learn α by taking into account the coherence of messages. When a_1 receives a message, it analyses which interpretations are allowed by the protocol and which are not. This information, however, is not definitive. An allowed word can be an incorrect interpretation for a received message, because protocols do not restrict completely all possible meanings. Moreover, a forbidden message can still be a correct interpretation; since some rules express relations between messages, a previously misinterpreted word (by the agent or by its interlocutor) can make a correct mapping appear impossible.

To handle this uncertainty, we propose a simple probabilistic learning technique. Agents maintain an *interpretation distribution* over possible meanings for foreign words, and update their values with the experience of interacting. Formally, for each $v_2 \in V_2$ that a_1 knows about (because it received it in a message at some point) and for each $v_1 \in V_1$, agent a_1 has a weight $\omega(v_2, v_1)$ that represents its confidence in that $\alpha(v_2) = v_1$. Using the bijectivity, we will keep $\sum_{v \in V_1} \omega(v_2, v) = 1$, but we do not require the same in the other direction since the foreign vocabulary is unknown a priori.

The technique that we propose can incorporate existing alignments, in the same way as it is done in [5] for the case when protocols are automata. These alignments represent a priori knowledge about the foreign vocabulary, that can have been obtained from previous experience, from a matching tool, or with other techniques such as a syntactic similarity measure. Since these techniques are never fully correct, previous alignments should not be trusted completely, and the techniques we propose can work as methods for repairing them. A previous alignment for agent a_1 is a partial function $\mathcal{A} : V_2' \times V_1 \to \mathbb{N}$, where V_2' is a set of terms. Note that $\mathcal{A}$ is not necessarily defined over V_2, since the vocabulary that a_2 uses may be unknown a priori. We interpret the confidences always positively, meaning that a mapping with low confidence has still more confidence than one that does not exist. If a_1 has a previous alignment $\mathcal{A}$, it initializes the interpretation distribution using that information. The agent first assigns raw values:

$$\omega(v_2, v_1) = \begin{cases} \mathcal{A}(v_2, v_1) & \text{if } (v_2, v_1) \in dom(\mathcal{A}) \\ 0 & \text{otherwise} \end{cases}$$

Then it normalizes the values using an exponential method (we use *softmax*) to start the learning with the values for all mappings in the interval $(0, 1)$, since the alignment can be wrong. If there is no previous alignment, the agent initializes the interpretation values with an uniform distribution, setting $\omega(v_2, v_1) = \frac{1}{|V_1|}$ for all $v_1 \in V_1$.

We explain in detail the learning techniques that we propose in Section 4. In the rest of this section we focus on the dynamics of the interaction: how agents choose which messages to say, finish the interaction, and decide interpretations for the messages they receive.

Choosing messages to send.

The choice of messages to utter is internal to each agent and depends on its interests while interacting. We do not impose any restriction on this, besides respecting the constraints in the protocol. Formally, a_1 can utter a message v_1 only if $\langle a_1, v_1 \rangle$ is a *possible message* as defined below.

DEFINITION 6. *Consider an agent a_1 following protocol $\mathfrak{P}_1 = \langle M_1, C_1, b \rangle$ and suppose interaction i has happened so far (and $i \models_p \mathfrak{P}_1$). The* possible messages *at that point are all messages $m \in M_1$ such that when they are performed the interaction remains a partial model, that is $i \cdot m \models_p \mathfrak{P}_1$.*

Finishing the interaction.

An interaction can finish in three situations. First, if there is a bound, reaching it implies the automatic end of the conversation. The interaction also finishes if an agent receives a message that has no possible interpretation. An agent can also finish the conversation whenever it wants if it considers that the interaction is *successful*, i.e., if it is a model of the protocol when it ends. In this case, it simply stops talking, producing a time-out that let the other agent realize the interaction finished.

Choosing interpretations.

When agent a_1 receives a message $v_2 \in V_2$ from a_2, it needs to interpret it in V_1 to be able to continue the interaction. To this aim, agents use the information in the interpretation distribution. Since agents assume there exists a bijective alignment, they always choose the same interpretation for a word during one interaction and they do not choose words that have already been chosen as interpretations.

Consider agent a_1 receives word v_2 after interaction i. Let $\mu : V_2 \to V_1$ be the *mappings made* function, where $\mu(v_2) = v_1$ if and only if v_1 was chosen as a mapping for v_2 before in the same interaction. The domain $dom(\mu)$ are the $v_2 \in V_2$ that a_1 received since the current task started, and its image $img(\mu)$ are the $v_1 \in V_1$ that were chosen as mappings. The set W of possible interpretations for v_2 is the set of words v_1 in V_1 such that $\langle a_2, v_1 \rangle$ is a possible message after i, and such that $v_1 \notin img(\mu)$. The inverse function $\mu^{-1} : V_1 \to V_2$ can be obtained since μ is bijective.

If $rnd(S)$ is a function that chooses an element randomly in the set S, the agent will choose an interpretation as follows:

$$\mu(v_2) = \begin{cases} \mu(v_2) & \text{if } v_2 \in dom(\mu) \wedge \mu(v_2) \in W \\ rnd(\operatorname{argmax}_{v_1 \in W} \omega(v_2, v_1)) & \text{if } v_2 \notin dom(\mu) \end{cases}$$

If either $v_2 \in dom(\mu)$ but $\mu(v_2) \notin W$, or $W = \emptyset$, the interaction is in one of the failure cases, and it finishes because a_1 stops talking.

4. LEARNING ALIGNMENTS FROM INTERACTIONS

To learn an alignment from the experience of interacting, agents make the following assumptions about their interlocutor. Essentially, these assumptions imply that the dynamics of the interaction described in the previous section are respected.

1. **Compliance:** An agent will not utter a message that violates the constraints, or that makes it impossible to finish successfully the interaction in the steps determined by the bound (or in finite steps if there is no bound).

2. **Non Abandonment:** An agent will not finish intentionally the conversation unless the constraints are fulfilled.

Agents learn when they decide how to interpret a received word. The overall method is simple: if a_1 receives v_2, it updates the value of $\omega(v_2, v_1)$ for all the words v_1 in a set $U \subseteq V_1$. The set U, unlike the set of possible interpretations W, can have both possible and impossible messages, and this classification is used to update the value in the interpretation distribution for a word.

A first decision concerns how to choose the set U. Since checking satisfiability is computationally expensive, this can impact considerably the overall performance of the methods. A first option is to update the value of all possible words ($U = V_1$), which can be slow for large vocabularies. Another option is to use only the possibilities that are considered until an interpretation is chosen. In this case, if a word v_2 is received and v_1 is chosen as an interpretation (because it is the possible word with largest ω for v_2), then $U = \{v \in V_1 \text{ such that } \omega(v_2, v) > \omega(v_2, v_1)\}$. A third possibility is a midpoint between these two, in which $U = \{v \in V_1 \text{ such that } \omega(v_2, v) \geq \omega(v_2, v_1)\}$. This option updates also the words that have the same value as the chosen interpretation. If the distribution is uniform when the learning starts, this updates many possible interpretations in the beginning and fewer when there is more information. This is the approach we choose, and the one used in the experimentation.

We present two methods to update the values in the interpretation distribution. The first one is a general technique that only takes into account whether messages are possible. This method is not exclusively designed for protocols that use LTL, and can be used for constraints specified in any logic for which agents have a satisfiability procedure. In a second method we show how the particular semantics of the protocols we introduced can be taken into account to improve the learning process.

4.1 Simple Strategy

The first approach consists in updating the value of interpretations for a received message, according to whether it is coherent or not with the protocol and the performed interaction. When a new message v_2 is received, the values for all interpretations are initialized as described in the last section. We use a simple learning method in which updates are a punishment or reward that is proportional to the value to update. We chose this method because it incorporates naturally previous alignments, it is very simple to formulate, and the proportional part updated can be manipulated easily to consider different situations, something that will be relevant for the second technique. However, it is not necessarily the most efficient choice, and other techniques, such as recording a history, can converge faster. Consider a reward and a punishment parameter $r_r, r_p \in [0, 1]$. We use the notion of *possible messages* in Definition 6 and update the values for $v_1 \in U$ as follows:

$$\omega(v_2, v_1) := \begin{cases} \omega(v_2, v_1) + r_r \cdot \omega(v_2, v_1) & \text{if } \langle a_2, v_1 \rangle \text{ is possible} \\ \omega(v_2, v_1) - r_p \cdot \omega(v_2, v_1) & \text{otherwise} \end{cases}$$

After all the updates for a given message are made, the values are normalised in such a way that $\sum_{v \in V_1} \omega(v_2, v) = 1$.

4.2 Reasoning Strategy

The simple strategy does not make use of some easily available information regarding which constraints were violated when a message is considered impossible. To illustrate the importance of this information, consider an interaction in which the Customer said water but the Waiter interpreted it as vino. If the Customer says beer, the Waiter may think that interpreting it as birra is impossible, since ordering two alcoholic beverages is not allowed (!$correlation($birra, vino$)$ would be violated). However, this is only due to having misunderstood water in the first place. In the reasoning approach, agents make use of the semantics of violated rules to perform a more fine-grained update of the values.

Before presenting the method, we need to divide constraints in two kinds, that will determine when agents can learn from them. Let us first introduce some new notation. We use the notion of *model* also for individual constraints ($i \models c$), and we say an interaction *violates* a constraint when $i \not\models c$. Given an interaction i and an interpretation v_1 for a received message, we say v_1 violates c if $i \models c$ but $i \,.\, \langle a_2, v_1 \rangle \not\models c$.

DEFINITION 7. *A constraint c is* semantically monotonic *if $i \models c$ implies that also all its extensions are models of c. A constraint c is semantically* non-monotonic *if there exist interactions i and i' such that i is a prefix of i', and $i \models c$ but $i' \not\models c$.*

The following proposition divides the constraints in our protocols between monotonic and non-monotonic.

PROPOSITION 1. *The constraints existence, correlation and response are semantically monotonic. All the rest of the constraints defined are semantically non-monotonic.*

PROOF (SKETCH) 1. *To prove non-monotonicity, it is enough to show an example that violates the constraint. For example, consider $before(m, m')$ and suppose an interaction i such that $m \notin i$ and $m' \notin i$. Then $i \models before(m, m')$ and $i \,.\, m' \not\models before(m, m')$. Monotonicity can be proven by reductio ad absurdum, showing that for any constraint c, if $i \,.\, m' \not\models c$ then necessarily $i \not\models c$. For example, if $i \,.\, m$ violates $existence(m', n)$, there are two options, either $m = m'$ or $m \neq m'$. Let $\#(m, i)$ and $\#(m, i \,.\, m')$ be the number of occurrences of c in i and $i \,.\, m'$ respective. In both cases, $\#(m, i \,.\, m') \geq \#(m, i)$, so if $\#(m, i \,.\, m') \leq n$, then necessarily $\#(m, i) \leq n$.*

Non-monotonic constraints can be used to learn while interacting, using the principle of compliance (if a constraint is violated, something must be wrong in the interpretations). Monotonic constraints could be used when the interaction ends, using the principle of non-abandonment (if not all constraints are satisfied, there must be an error). However, since our agents cannot communicate in any way, they do not know if their interlocutor considers the constraints were satisfied or not, difficulting the decision of how to update values. In this work we focus on handling non-monotonic constraints, leaving monotonic ones for future work where an ending signal is introduced.

If a_1 decides that v_1 is not a possible interpretation for v_2, let $Viol$ be all $c \in C_1$ that are violated by v_1. It is possible that $Viol = \emptyset$, when it is not one but a set of constraints that becomes unsatisfiable, or when the message makes the

Violated Constraint	Action
-	$\omega(v_2, v_1) := \omega(v_2, v_1) - r_p \cdot \omega(v_2, v_1)$
$absence(\langle a_2, v_1 \rangle, n)$	$\omega(v_2, v_1) := 0$
$!correlation(\langle a_2, v_1' \rangle, \langle a_2, v_1 \rangle)$ $!response(\langle a_2, v_1' \rangle, \langle a_2, v_1 \rangle)$ $!before(\langle a_2, v_1' \rangle, \langle a_2, v_1 \rangle)$ $!premise(\langle a_2, v_1' \rangle, \langle a_2, v_1 \rangle)$ $!imm\_after(\langle a_2, v_1' \rangle, \langle a_2, v_1 \rangle)$ $imm\_after(\langle a_2, v_1' \rangle, \langle a_2, v_1'' \rangle)$	$\omega(v_2, v_1) := \omega(v_2, v_1) - r_p \cdot \omega(\mu^{-1}(v_1'), v_1')$ $\omega(\mu^{-1}(v_1'), v_1') := \omega(\mu^{-1}(v_1'), v_1') - r_p \cdot \omega(v_2, v_1)$
$premise(\langle a_2, v_1' \rangle, \langle a_2, v_1 \rangle)$	$\omega(v_2, v_1) := \omega(v_2, v_1) - r_p \cdot \omega(\mu^{-1}(v_1''), v_1'')$ $\omega(\mu^{-1}(v_1''), v_1'') := \omega(\mu^{-1}(v_1''), v_1'') - r_p \cdot \omega(v_2, v_1)$ where v_1'' is the last word in the interaction
$before(\langle a_2, v_1' \rangle, \langle a_2, v_1 \rangle)$ $relation(\langle a_1, v_1' \rangle, \langle a_2, v_1 \rangle)$	$\omega(v_2, v_1) := \omega(v_2, v_1) - r_p \cdot \omega(v_2, v_1)$

Table 2: Updates for each violated constraint
(*relation* stands for any relational constraint)

interaction impossible to finish successfully in the number of utterances indicated by the bound.

To explain the update in the reasoning strategy, consider again two reward and punishment parameters $r_r, r_p \in [0, 1]$ respectively. If an interpretation v_1 is possible for a received message v_2, then $\omega(v_2, v_1) := \omega(v_2, v_1) + r_r \cdot \omega(v_2, v_1)$. If it is not, it is updated as indicated in Table 2, according to which rule was violated. These actions are performed iteratively for each broken rule. After the updates, all values are normalized to obtain $\sum_{v \in V_1} \omega(v_2, v) = 1$. In this case, we need to use a normalization method that maintains the values that equal 0.

Let us explain the motivation under each update. Violating the existential non-monotonic constraint is different to violating a relation constraint, because it expresses a condition over only one action. Thefore if the chosen interpretation violates it, it must be wrong. For this reason, the value of the mapping is set to 0.

When a negative relational constraint is violated, any of the two messages it relates can have been misinterpreted. The update we present here is inspired by Jeffrey's rule of conditioning [18], that states that if Q is a probability distribution for a partition $E_1 \ldots E_n$ of possible evidence, then the posterior probability of A is computed as $Q(A) = \sum_{i=0}^{n} P(A|E_i) \cdot Q(E_i)$.

Let v_2' be a received message such that $\mu(v_2') = v_1'$. Suppose a_1 receives a new message v_2, and it discovers that mapping it with v_1 violates a constraint $!relation(\langle a_2, v_1' \rangle, \langle a_2, v_1 \rangle)$ (where $!relation$ is any negative relational constraint). This means that $P(\alpha(v_2) = v_1 \wedge \alpha(v_2') = v_1') = 0$, and therefore $P(\alpha(v_2) = v_1 | \alpha(v_2') = v_1') = 0$. Then, if Pos is the set of possible mappings, we can write:

$$\omega(v_2, v_1) := \omega(v_2, v_1) \cdot \sum_{v_1' \in Pos} \omega(v_2', v_1')$$

or, what is equivalent, if $Impos$ is the set of incompatible mappings, since $\sum_{v \in v_1} \omega(v_2, v_1) = 1$:

$$\omega(v_2, v_1) := \omega(v_2, v_1) - \sum_{v_1' \in Impos} \omega(v_2', v_1')$$

We implement this by subtracting for each mapping a value that is proportional to the confidence in the other mapping. Since agents do not explore all the possible interpretations, the value is only approximated considering only the interpretations that have more confidence.

In the case of imm_after being violated, the constraint will be between a previously matched word v_1' and a new word v_1'', but the update is identic. For *premise*, the interpretation depends on having interpreted correctly the message immediatly before. If *before* is violated, there is not one unique message to take into account, so agents use the default punishment r_p. The same occurs when $Viol = \emptyset$, and when a violated constraint depends on a message that a_1 sent.

Lastly, let us discuss the value of r_p. Choosing $r_p = \frac{1}{|V_1|}$, the technique has the effect of subtracting a larger value when the agent is confident in the correctness of the previous interpretation. This is the value that we use.

5. EXPERIMENTAL EVALUATION

Before presenting the results of experiments that show the performance of the proposed techniques, let us discuss the generation of data for experimentation.

Due to the lack of existing datasets, we performed all the experimentation with randomly generated data. We created protocols with different sizes of vocabulary and of set of constraints. To generate a protocol from a vocabulary, we randomly chose constraints and added them to the protocol if it remained satisfiable with the new rule. In the implementation, we express constraints over words instead of over messages, so a constraint are valid for any combination of senders. This makes the experimentation faster since protocols are smaller. We used bounds for performance reasons, that in all cases had the value of the vocabulary size. We created pairs of compatible protocols $\mathfrak{P}_1, \mathfrak{P}_2$, with vocabular-
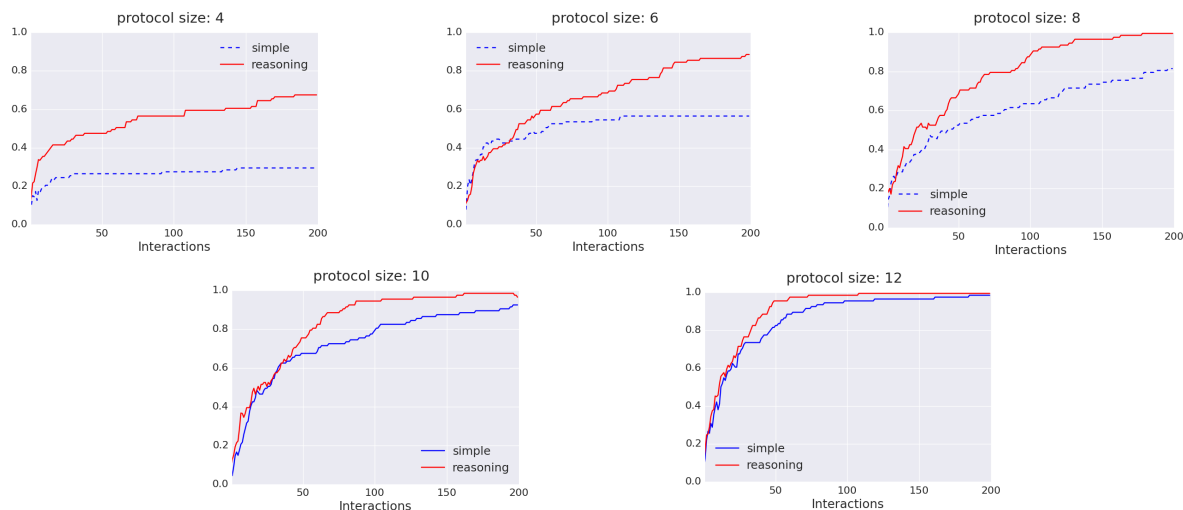
Figure 2: Results for a vocabulary of size 10

ies V_1 and V_2 respectively by building a bijective alignment $\alpha : V_2 \to V_1$ and then using it to translate the constraints in a protocol $\mathfrak{P}_2$, obtaining $\mathfrak{P}_1$. We used the NuSMV model checker [6] to perform all the necessary satisfiability checks.

To evaluate how agents learn α from the experience of interacting, we first need to define a measure of how well an agent knows an alignment. Since agents always choose the possible mapping with highest weight, we can easily extract from the interpretation distribution an alignment that represents the first interpretation choice for each foreign word in the following interaction. This alignment, that we call α_i for agent a_i, is one of the mappings with highest weight for each foreign word. Formally, for a_1, the domain of α_1 is $v_2 \in V_2$ such that $\omega(v_2, v)$ is defined for all $v \in V_1$, and $\alpha_1(v_2) = rnd(\text{argmax}_{v \in V_1} \omega(v_2, v))$. Note that if there are multiple possibilities with the same value, $\alpha_1(v_2)$ takes a random word between them.

To compare this alignment with α, we used the standard precision and recall measures, and their harmonic mean combination, commonly known as *F-score*. To use the standard definitions directly, we only need to consider alignments as relations instead of functions: given an alignment $\alpha : V_2 \to V_1$, the corresponding relation is the set of pairs $\langle v_2, v_1 \rangle$ such that $\alpha(v_2) = v_1$. Given two alignments β and γ expressed as relations, the *precision* of β with respect to γ is the fraction of the mappings in β that are also in γ:

$$precision(\beta, \gamma) = \frac{\mid \beta \cap \gamma \mid}{\mid \beta \mid}$$

while its *recall* is the fraction of the mappings in γ that were found by β:

$$recall(\beta, \gamma) = \frac{\mid \beta \cap \gamma \mid}{\mid \gamma \mid}$$

The *F-score* of β is computed as follows:

$$F - score(\beta, \gamma) = 2 \cdot \frac{precision(\beta, \gamma) \cdot recall(\beta, \gamma)}{precision(\beta, \gamma) + recall(\beta, \gamma)}$$

We performed experiments parametrized with a protocol and vocabulary size. In all experiments, agents are sequentially given pairs of compatible protocols, and after each interaction we measure the F-score of their alignments with respect to α. The same protocols can appear repeatedly in the sequence, but eventually a new one always appear, so at some point there is one unique alignment under which they are all compatible. We first intended to perform experiments with the same vocabulary sizes that are used for testing in [1], which are $5, 10, 15, 40, 80$. However, the interactions for size 80 were too slow to be able to perform a reasonable amount of repetitions for each technique. This problem is intrinsic to the protocols we use (since agents have to decide if the messages they want to send are possible), and should be taken into account in future work. Since varying the vocabulary size did not provide particularly interesting results, we show here the experiments for a vocabulary of 10 words and four different protocol sizes. We used punishment and rewards parameters of $r_p, r_r = 0.3$ for the simple strategy. Each experiment was repeated 10 times, and we averaged the results.

In a first experiment, we let agents interact 200 times and measured the way they learned the alignment for different protocol sizes, shown in Figure 2. The *F-score* if computed by averaging the *F-score* of each agent that participates in the interaction. The curves show that the reasoning strategy is always better than the simple one, but this difference is more dramatic when the protocols are smaller. This is because there is less information, so using it intelligently makes a great difference. The curves are sharp in the beginning, when agents do not have much information about α, and they make many mistakes from which they learn fast. When agents reach a reasonable level of precision and recall, mistakes are less frequent, and therefore the pace of the learning slows considerably. Although this affects the convergence to an F-score of 1.0, it also means that, after a certain number of interactions, agents have learn enough to communicate successfully most of the times. The sparse mistakes that they will make in future conversations make them learn more slowly the remaining mappings. Table 3 shows how fast agents reach a reasonable level of F-score, that we chose of 0.8 based on [10].

In a second experiment, we studied the performance of our methods when used to repair existing alignments. To this

	6	8	10	12
simple	-	187	101	47
reasoning	139	87	57	33

Table 3: 0.8 convergence

aim, we created alignments with different values of precision and recall with respect to α, that we divided into three categories. Low quality alignments had precision and recall 0.2, medium quality 0.5, and high quality 0.8. We only studied alignments with the same value of precision and recall to reduce the number of combinations and have a clear division of quality levels. We made agents with this alignments interact, using the simple and the reasoning technique. The results can be seen in Figure 3. The number next to each agent class in the reference is the precision and recall of the alignment that was given to that agent. Again, the reasoning strategy performs particularly better when only little information is available; in this case, the repair is much better than when using the simple strategy for alignments of low quality.
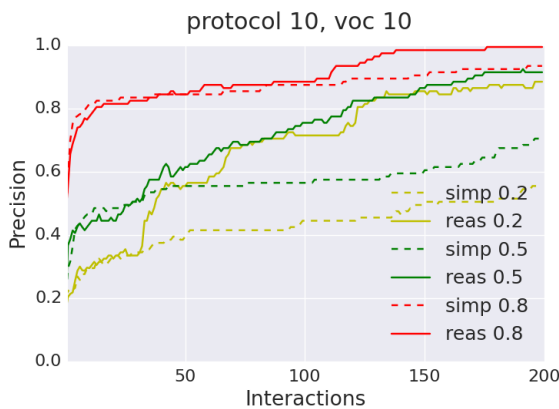

Figure 3: Results for different alignment qualities

6. RELATED WORK

As we already mentioned, a large portion of the existent work that tackles the problem of vocabulary alignment for multi-agent communication consists of techniques to let agents discuss and agree on a common vocabulary or alignment. Some of these approaches use argumentation or negotiation techniques [16, 12], performing an offline negotiation that takes place before actual interactions can start. Other approaches, such as the three-level schema developed by van Diggelen [8], discuss the meaning of works on a by-demand basis, that is, only when it is necessary to use them to perform some task.

Other approaches use different kinds of grounding to learn alignments or the meaning of words from interactions. The well established research by Steels [21] considers a situation in which learners share a physical environment, and have the ability of pointing to things to communicate if they are talking about the same things. Goldman et al. [11], investigated how agents can learn to communicate in a way that max-

imises rewards in an environment that can be modelled as a Markov Decision Process. In [4], the authors study a version of the multiagent, multiarmed bandit problem in which agents can communicate between each other with a common language, but message interpretations are not know. Closer to our approach is the work by Euzenat on cultural alignment repair [9], and particularly the already mentioned work by Atencia and Schorlemmer [1], where the authors consider agents communicating with interaction protocols represented with finite state automata, and use a shared notion of task success as grounding. Our work, instead, considers only the coherence of the interaction as grounding. To the best of our knowledge, the work presented in this paper is the first approach in which agents learn languages dynamically from temporal interaction rules, taking into account only the coherence of the utterances.

A very well studied problem consists on learning different structures from experience, such as grammars [7] or norms [17]. Our approach can be seen as the reverse of these approaches, where some structure is considered shared and used to learn from.

7. CONCLUSIONS AND FUTURE WORK

The techniques we propose allow agents to learn alignments between their vocabularies only by interacting, assuming that their protocols share the same set of models, without requiring any common meta-language or procedure. We made a number of assumptions, namely requiring agents to share the same models for each task and considering the existence of a bijective alignment. These assumptions can be relaxed, but at the cost of a slower learning, since an extra component of uncertainty is added. We also assumed a perfect interactions setting in which no message is lost. Although our methods converge slowly, they use only very general information, and can be easily combined with other methods that provide more information about possible mappings, as we show with the integration of previous alignments.

There exist many possible directions of research derived form this work. First, we could relax the assumption that agents do not share any meta-language, considering agents that can in some way exchange information about the interaction. For example, considering only that agents can communicate whether they finished a task successfully would make possible to reason about monotonic rules when an interaction ends. An approach like this would relate our work to the one by Santos [16] about dialogues for meaning negotiation. Another possible extension considers agents that prioritize the language learning to performing the task. In this case, agents would have to decide what utterances it is more convenient to make in order to get more information about the alignment. An aspect that should be improved in future work is the performance in terms of runtime per interaction, since it was very slow for larger vocabularies.

Acknowledgments

REFERENCES

[1] M. Atencia and M. Schorlemmer. An interaction-based approach to semantic alignment. *Journal of Web Semantics*, 12-13:131–147, 2012.

[2] M. Baldoni, C. Baroglio, and E. Marengo. Behavior-oriented commitment-based protocols. In *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, pages 137–142, 2010.

[3] M. Baldoni, C. Baroglio, E. Marengo, and V. Patti. Constitutive and regulative specifications of commitment protocols: A decoupled approach (extended abstract). In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 4143–4147, 2015.

[4] S. Barrett, N. Agmon, N. Hazon, S. Kraus, and P. Stone. Communicating with unknown teammates. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *ECAI 2014 — 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic — Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 45–50. IOS Press, 2014.

[5] P. Chocron and M. Schorlemmer. Attuning ontology alignments to semantically heterogeneous multi-agent interactions. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, The Hague, The Netherlands*, pages 871–879, 2016.

[6] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In *International Conference on Computer Aided Verification*, pages 359–364. Springer, 2002.

[7] C. de la Higuera. *Grammatical Inference: Learning Automata and Grammars.* Cambridge University Press, New York, NY, USA, 2010.

[8] J. V. Diggelen, R. Beun, F. Dignum, R. M. V. Eijk, and J. J. Meyer. Ontology negotiation: Goals, requirements and implementation. *Int. J. Agent-Oriented Softw. Eng.*, 1(1):63–90, Apr. 2007.

[9] J. Euzenat. First Experiments in Cultural Alignment Repair. In *Semantic Web: ESWC 2014 Satellite Events*, volume 8798, pages 115–130, 2014.

[10] J. Euzenat, C. Meilicke, H. Stuckenschmidt, P. Shvaiko, and C. Trojahn. Ontology alignment evaluation initiative: Six years of experience. *Journal on Data Semantics XV*, pages 158–192, 2011.

[11] C. V. Goldman, M. Allen, and S. Zilberstein. Learning to communicate in a decentralized environment. *Autonomous Agents and Multi-Agent Systems*, 15(1):47–90, Aug. 2007.

[12] L. Laera, I. Blacoe, V. A. M. Tamma, T. R. Payne, J. Euzenat, and T. J. M. Bench-Capon. Argumentation over ontology correspondences in MAS. In E. H. Durfee, M. Yokoo, M. N. Huhns, and O. Shehory, editors, *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*, page 228, 2007.

[13] E. Marengo, M. Baldoni, C. Baroglio, A. K. Chopra, V. Patti, and M. P. Singh. Commitments with regulations: Reasoning about safety and control in regula. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pages 467–474, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[14] M. Montali. *Specification and verification of declarative open interaction models: A logic-based approach*, volume 56 of *Lecture Notes in Business Information Processing*. Springer-Verlag New York Inc, 2010.

[15] M. Pesic and W. M. P. van der Aalst. A declarative approach for flexible business processes management. In *Proceedings of the International Conference on Business Process Management Workshops*, BPM, pages 169–180, Berlin, Heidelberg, 2006. Springer-Verlag.

[16] G. Santos, V. Tamma, T. R. Payne, and F. Grasso. A dialogue protocol to support meaning negotiation. (extended abstract). In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '16, pages 1367–1368, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

[17] S. Sen and S. Airiau. Emergence of norms through social learning. In *IJCAI*, volume 1507, page 1512, 2007.

[18] G. Shafer. Jeffrey's rule of conditioning. *Philosophy of Science*, 48(3):337–362, 1981.

[19] N. Silva, G. I. Ipp, and P. Maio. An approach to ontology mapping negotiation. In *Proceedings of the Workshop on Integrating Ontologies*, pages 54–60, 2005.

[20] M. P. Singh. A social semantics for agent communication languages. In *Issues in Agent Communication*, pages 31–45, London, UK, 2000. Springer-Verlag.

[21] L. Steels. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):169–194, Oct. 1998.