

Vocabulary Alignment in Openly Specified Interactions

Paula Chocron

*Artificial Intelligence Research Institute, IIIA-CSIC
Bellaterra (Barcelona), Catalonia, Spain*

*Universitat Autònoma de Barcelona
Bellaterra (Barcelona), Catalonia, Spain*

PAULACHOCRON@GMAIL.COM

Marco Schorlemmer

*Artificial Intelligence Research Institute, IIIA-CSIC
Bellaterra (Barcelona), Catalonia, Spain*

MARCO@IIIA.CSIC.ES

Abstract

The problem of achieving common understanding between agents that use different vocabularies has been mainly addressed by techniques that assume the existence of shared external elements, such as a meta-language or a physical environment. In this article, we consider agents that use different vocabularies and only share knowledge of how to perform a task, given by the specification of an interaction protocol. 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 their performance experimentally.

1. Introduction

Multiagent systems are conceived as open, distributed and heterogeneous environments where agents from different design backgrounds interact, each of them contributing with their own knowledge or abilities. From autonomous cars that inform each other of their position and other traffic-related data to improve circulation and avoid accidents, to medical assistant robots who collaborate with humans in hospitals, the multiagent scenarios envisioned by the artificial intelligence community require communication to coordinate their actions when the objective is to perform a task together.

Agent heterogeneity is commonly addressed by means of some mediating architectural abstraction such as, for instance, the A&A meta-model for multiagent systems (Omicini et al., 2008). By means of *boundary agents* one defines interfaces between agents and the multiagent interaction space so as to deal with all sources of change in a uniform way (Mariani & Omicini, 2015). In very dynamic settings, however, it will be difficult to ensure that all participants use the same vocabulary in their communication, and that they interpret it equivalently; or, given different vocabularies, to ensure that these are translated by means of translator artifacts in a uniform way. Even if this were possible at some point in time, vocabularies may continuously change in an evolving system, when they are adapted to satisfy new needs, for instance. In these situations, it becomes essential to address the problem of *dynamic* heterogeneity: finding ways in which agents in an open, distributed and heterogeneous multiagent system may be capable of dynamically aligning their respective

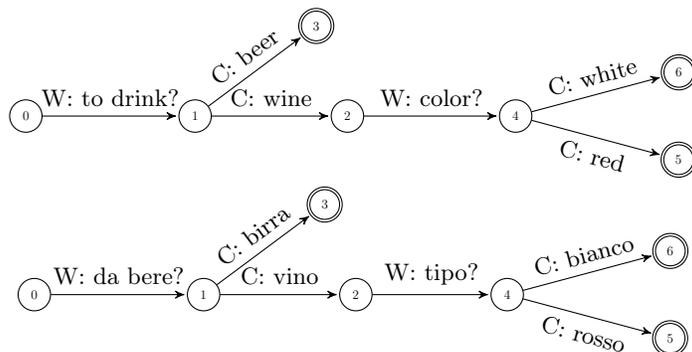


Figure 1: FSM protocols for Waiter (W) and Customer (C)

vocabularies to successfully interact through communication, even when confronted with continuous change and evolution. This is necessary to ensure meaningful interactions, and therefore crucial for the success of systems in which agents act jointly by communicating.

The multiagent community has proposed several solutions to the problem of vocabulary alignment in multiagent systems. As we will discuss in Section 2, in general these contributions either consider the existence of external *contextual* elements that all agents perceive, or assume agents share a common meta-language that they can use to discuss the meaning of words and their alignments. In this article we take a different approach to multiagent vocabulary alignment, proposing a method that uses only information about the interaction that is being performed. An analogy with human communication can be useful to introduce this idea. Consider an English speaker trying to get a glass of wine in Italy. While the first time the situation can be confusing, after some attempts the customer will likely learn how to perform the interaction successfully in Italian. This is, in part, due to two factors. First, the customer and the seller (mostly) agree on how the interaction for buying wine should unfold. Second, the interaction context is very reduced, which means that the number of possible mappings can be kept small, and therefore the correct ones are easier to discover.

To make the example more concrete, consider the interaction in Figure 1, in which an English speaking customer C communicates with an Italian waiter W . Although W , may not “understand” what C wants to drink, it could just guess one of the possible message alignments and attempt to continue its task. After repeated interaction with English speaking customers, W would be capable to progressively learn which alignments lead to successful outcomes, relating, for example, *beer* to *birra*. This idea was first developed by Atencia and Schorlemmer (2012)—and further extended by Chocron and Schorlemmer (2016)— who consider an interaction as successful when both agents jointly reach a final state in the execution of their respective interaction protocols. However, their use of 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, in the approach presented by Atencia and Schorlemmer (2012), agents learn an alignment that is only useful for one task. How this can be extrapolated to other interactions is not explored.

In our previous work (Chocron & Schorlemmer, 2017b) we presented a version of the interaction-based alignment technique for agents that specify interactions with constraint-

based *open protocols* that define temporal rules about what can be said, instead of forcing particular executions (Endriss, 2006; Pham & Harland, 2007). For example, instead of specifying that the waiter must ask *color?* after the customer asks for wine, these protocols could include a constraint saying that the color must be asked at some point in time after wine is ordered, leaving to the agent the decision of exactly when to do it. These protocols provide a much more flexible specification that does not define the complete flow of the messages but only some relevant constraints, avoiding over-specification. In particular, we used ConDec protocols (Pesic & van der Aalst, 2006; Pesic et al., 2007), that employ linear temporal logic to specify constraints. One ConDec protocol may not be enough to completely define the meaning of all words in a vocabulary, so in previous work (Chocron & Schorlemmer, 2017b) we study how agents can learn mappings when they perform different tasks.

Learning while interacting has several advantages. First of all, no extra interactions or information are necessary; agents exchange only the messages they need to perform the task. This implies that they do not need to access extra information in the semantic representations of their interlocutors, which might be impossible or undesirable. Since agents learn gradually as different interaction opportunities appear, they can use what they learned earlier, even if they still do not know the alignment completely. A more subtle advantage is that agents learn the vocabulary that is actually used to perform the tasks they need to perform. This is particularly important when they are restricted to a specific domain, that can have its particular jargon.

Our techniques do require that agents can participate in many interactions, particularly when vocabularies are large. Nowadays, however, obtaining these rehearsal interactions is likely cheaper than crafting a linguistic or ontological resource that is useful to obtain alignments. In particular, these interactions do not need to have any type of human intervention. For example, consider an agent trying to learn a new vocabulary that is already spoken by other artificial interlocutors. This agent can be introduced into the new community and start learning immediately. It is only necessary to identify it as a learner so that other agents are aware of its situation. A different application scenario that we envision for our techniques is one in which there are emergent semantics (Aberer et al., 2003). This term refers to new meaning that is created dynamically by other interlocutors, which is difficult to capture since it is usually not explicit. Using our techniques, an artificial agent could infer the meaning of expressions that appear spontaneously in social media communities only by analyzing their use. Since the agent only needs to learn the meaning of a small vocabulary, it does not require that many interactions.

In this article we extend and improve the work presented in our previous work (Chocron & Schorlemmer, 2017b). Concretely, our contributions are:

- We present a formalization of the learning process (Section 5).
- We propose new learning methods. Concretely, we introduce agents that choose their utterances to improve their chances of learning (Section 5).
- We improve the learning techniques, obtaining agents that learn significantly faster (Section 5 and 6).
- We expand the experiments, considering new interesting cases (Section 6).

- We compare our techniques with an existing method (Section 6).
- We discuss how a particular class of rules (called *p-necessary*) can be taken into account in the learning process (Section 7).
- We compare the agents we propose, that use a probabilistic learning technique, with an agent that makes only logical inferences (Section 8).

The remainder of this article is organized as follows. After reviewing related work in Section 2, we introduce *open interaction protocols* in Section 3, and in Section 4 we define a framework for interacting with partners that use different vocabularies. Section 5 presents different techniques to learn an alignment from the experience of interacting that improve the ones in previous work (Chocron & Schorlemmer, 2017b), as well as other ones that consider agents that actively want to learn. All 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 the different techniques experimentally in Section 6. Section 7 discusses agents that can use a particular class of *p-necessary* constraints in the learning process, and in Section 8 we compare our probabilistic learning technique to one based on logical inference. We conclude with Section 9.

2. Related Work

Vocabulary alignment has been extensively studied in the field of ontology and ontology matching (Kalfoglou & Schorlemmer, 2003; Euzenat & Shvaiko, 2013), but the dynamic character of multiagent interactions adds a dimension of heterogeneity that cannot be captured solely with the static knowledge that ontologies and ontology alignments provide about some particular domain. Consequently, the multiagent community has explored the vocabulary alignment problem also in relationship to the dynamics of multiagent interactions.

Steels (1998) explored mechanisms by which a group of agents forms a vocabulary and ontology in a self-organised way through local interaction and without central control authority. His focus was on robotic agents that share the same sensorimotor capabilities and which, by means of language games, develop their own shared ontology, creating and grounding their vocabulary on the perception of a shared physical space.

Reaching a common vocabulary or vocabulary alignment has also been the focus of the work of Bailin and Truszkowski (2002), Laera et al. (2007), Santos et al. (2016), van Diggelen et al. (2006), and Silva et al. (2005), but without assuming the existence of external elements that all agents perceive, drawing instead from techniques developed for multiagent argumentation and negotiation. Without sharing the same physical space, agents can now communicate remotely but they need a common meta-language to reach agreements over word meanings and their alignments.

The idea of letting agents create and refine vocabulary alignments by means of interaction games in the spirit of Steels (1998) has also been explored by Euzenat (2014, 2017). His approach assumes a shared collection of objects with shared Boolean features that are to be classified according to the diverse vocabularies. Interaction games are successful if the resulting classification of an object by both agents is compatible with the public ontology alignment. Otherwise these alignments are repaired using adaptation operators.

Sharing the instances that are classified into different categories is also the assumption behind the work of Wang and Gasser (2002), who present an alignment algorithm of ontologies in a multiagent system based on the k-means algorithm, a classical partition-based clustering method. Ontologies are partitions of domain instances into categories, and the alignment is done by communicating instances between two agents.

Focusing on the information one can gain from the interactions themselves, Besana and Robertson (2007) have explored how interactions between agents can be used to predict vocabulary alignments, reducing the search space of a priori possible mappings between vocabularies and without the need to map the entire ontologies, which simplifies the ontology mapping problem for open and distributed environments.

A complementary angle to the vocabulary alignment problem in multiagent systems has been pursued by Bravo and Velazquez (2008) who focused on pragmatic interoperability, i.e., the actual use of exchanged messages and their effect according to the interaction dynamics as specified in multiagent interactions. The pragmatic similarity they define and compute bears some resemblance with the sort of alignment that one can infer from what Atencia and Schorlemmer (2007) called the ‘communication product’, which captures the global interaction with respect to the compatible message-passing behaviour of agents with different interaction models.

Protocol compatibility is also behind the techniques developed by Ancona et al. (2018) for computing the vocabulary alignments that would allow an agent designed for a particular multiagent protocol to be successfully deployed in a different one. Protocols are specified as trace expressions, which are more expressive than the FSA-based formalism studied by Bravo and Velazquez (2008) and Atencia and Schorlemmer (2007). But as with the ‘communication product’, access to the entire protocol specifications of the agents is needed in order to compute a mapping, which needs to be established before the agents start interacting.

Goldman et al. (2007) investigated how agents can learn to communicate in a way that maximizes rewards in an environment that can be modeled as a Markov Decision Process. Barrett et al. (2014) studied a version of the multiagent, multiarmed bandit problem in which agents can communicate with each other in a common language, but message interpretations are not known.

In the area of cognitive language acquisition there exists extensive work on computational models of first language acquisition that models the way in which children learn word-meaning mappings from disambiguating sentences that are described in a given observable situation (Xu & Tenenbaum, 2007; Siskind, 1996; Angluin & Becerra-Bonache, 2017; Akhtar & Montague, 1999). However, this implies that there exists a physical situation that is observed by both the learner and the speaker. The problem of learning translations only from the experience of interacting is therefore notably unexplored also in this area. Our work could also be a model of second language immersion learning that takes advantage of the cultural procedural similarities between the speakers.

How to achieve agent coordination from interaction experience, by learning from repeated interaction with multiple agents, has been studied for instance for norm emergence by Sen and Airiau (2007); and how to infer coordination structures such as finite automata or grammars has also been described by de la Higuera (2010). These approaches, in which the vocabulary is shared while some other coordination structure —either norms or grammars— is inferred, can be seen as the reverse of our work.

3. Open Interaction Protocols

To explain how agents can infer alignments from the information about the dynamics of an interaction, it is first necessary to define how this information is represented. To this end, we use *interaction protocols* that restrict what can or cannot be said by an agent at a certain time. The question of what should be expressed by an interaction protocol has been extensively discussed in the multiagent systems community. Traditional approaches such as finite state automata and Petri Nets are simple to read and verify, but extremely rigid, since any message that can be uttered at a certain time needs to be explicitly defined. This lack of flexibility hinders their design, change, or reuse, making them only adequate for simple and static interactions.

A different approach, that provides more flexibility, consists in specifying protocols by constraining 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 (2006) as a language to describe business protocols. In later work they have been used as a specification language for agent interactions, for example by Montali (2010), who presents an extension of the ConDec language and tools for its verification, and by Baldoni et al. (2010, 2013), who integrates ConDec constraints with commitment protocols, a framework to specify interactions with social semantics (Singh, 2000). An important advantage of ConDec protocols is that they use linear temporal logic, a well-known formalization for which many reasoning tools are available. These protocols specify only the temporal aspect of the interaction, without any link to a more global view of the task that is being performed (see the work by Marengo et al. 2011 for a discussion on this), or to any other semantic information about the concepts that agents are talking about. In this article we show that temporal information is enough to infer an alignment between vocabularies that allows agents to communicate meaningfully. Of course, agents with richer protocols can incorporate the extra information to the learning process to make it faster or more precise.

The rest of this section is divided in three parts. In Section 3.1, we present the basic notions of ConDec protocols as they were defined by Pesic and van der Aalst (2006). In Section 3.2 we present our adaptation to use ConDec protocols as specifications of interactions between agents that may use different vocabularies. In Section 3.3 we introduce the idea of *compatibility* between protocols.

3.1 Preliminaries

Linear Temporal Logic (LTL from now on) is a natural choice to express constraints about actions that occur in time. We will describe it following the conventions used in Huth and Ryan (2004). The syntax of LTL consists of a finite set of propositional variables \mathbf{Atoms} , the logical operators \neg , \wedge and \rightarrow , and the temporal modal operators $\{G, F, X, W\}$.¹ The set of LTL formulas over \mathbf{Atoms} is defined as follows:

1. if $p \in \mathbf{Atoms}$, then p is an LTL formula, and

1. This set is not minimal, but it is the most useful one for our purposes. For the same reason we use weak until (W) instead of until (U).

2. if ϕ, ψ are LTL formulas, then $\neg\phi$, $\phi \wedge \psi$, $\phi \rightarrow \psi$, $G\phi$, $F\phi$, $X\phi$, $\phi W \psi$ are LTL formulas.

LTL formulas are interpreted over sequences of states, with each state being associated to a truth-valuation of the propositional variables in \mathbf{Atoms} . The intuitive idea is that $G\phi$ means that ϕ must be true in the truth-valuation of all following states, $F\phi$ means that ϕ must be true eventually, $X\phi$ means that ϕ must be true in the next state, and $\phi W \psi$ means that ϕ must be true until ψ is true, if that ever happens, or true always otherwise. Formally, consider a set of states S , and a labeling function $L : S \rightarrow 2^{\mathbf{Atoms}}$ assigning each state to a subset of \mathbf{Atoms} . Given a propositional variable $p \in \mathbf{Atoms}$ and a state $s \in S$, the value of p in s is interpreted as **true** if $p \in L(s)$ and as **false** otherwise. A word $w = s_1, s_2, \dots$ is an infinite sequence of states in S , that is, an element in S^ω . We write w^i for the suffix starting at s_i , that is, $w^i = s_i, s_{i+1}, \dots$.

The semantics of LTL is defined as follows. Let $p \in \mathbf{Atoms}$, and let ϕ, ψ be two LTL formulas.

- $w \models p$ if $p \in L(s_1)$
- $w \models \neg\phi$ if $w \not\models \phi$
- $w \models \phi \wedge \psi$ if $w \models \phi$ and $w \models \psi$
- $w \models \phi \rightarrow \psi$ if $w \models \psi$ whenever $w \models \phi$
- $w \models G\phi$ if for all $i \geq 1$, $w^i \models \phi$
- $w \models F\phi$ if there is some $i \geq 1$ such that $w^i \models \phi$
- $w \models X\phi$ if $w^{2:} \models \phi$
- $w \models \phi W \psi$ if either there is some $i \geq 1$ such $w^i \models \psi$ and for all j such that $1 \leq j \leq i - 1$ we have $w^j \models \phi$, or for all $k \geq 1$ we have $w^k \models \phi$

A set of LTL formulas, called a *theory*, is *satisfiable* if there exists a sequence of states in S for which all the formulas are true. In that case, the sequence 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 sequence of states in S is a model of a theory.

The ConDec specification language provides a set of constraint templates, which rename some particular basic LTL formulas. The first column of Table 1 shows the ones that we use in this article. We use a simplified version² of the original templates that were introduced by Pesic and van der Aalst (2006) and by van der Aalst and Pesic (2007). The most similar version is the one proposed by Baldoni et al. (2010). A ConDec protocol, then, is a set of constraints over a particular set of variables.

2. Since we are not interested in the usability of the protocols here, we do not include those constraints that work as syntactic sugar, such as *exactly*(n, a), that can be replaced by including *existence*(n, a) and *!existence*(n, a).

Definition 1. (ConDec protocol) Let M be a set of propositional variables. We call $Cons(M)$ the set of all ConDec constraints over M , that is, the templates in the first column of Table 1 for any $m, m' \in M$ and $n \in \mathbb{N}^+$. A *ConDec protocol over M* is a finite subset of $Cons(M)$. ■

The LTL meanings of the constraints can be seen in the second column of Table 1. In general, constraints are classified into two types. Existential constraints (*existence* and *!existence*) predicate over how many times some action can be performed. Relational constraints (the remaining ones) describe binary relations between two actions. Constraints can be positive or negative, representing obligations or prohibitions respectively. Negative constraints start with an exclamation mark.

Constraint	LTL meaning
$existence(1, m)$	Fm
$existence(n + 1, m)$	$F(m \wedge Xexistence(n, m))$
$!existence(n, m)$	$\neg existence(n, m)$
$correlation(m, m')$	$Fm \rightarrow Fm'$
$!correlation(m, m')$	$Fm \rightarrow \neg Fm'$
$response(m, m')$	$G(m \rightarrow Fm')$
$!response(m, m')$	$G(m \rightarrow \neg Fm')$
$before(m, m')$	$\neg m' W m$
$!before(m, m')$	$G(Fm' \rightarrow \neg m)$
$premise(m, m')$	$G(Xm' \rightarrow m)$
$!premise(m, m')$	$G(Xm' \rightarrow \neg m)$
$imm_after(m, m')$	$G(m \rightarrow Xm')$
$!imm_after(m, m')$	$G(m \rightarrow X\neg m')$

Table 1: LTL definitions of constraints, where $m, m' \in M$ and $n \in \mathbb{N}^+$.

3.2 Open Protocols as Interaction Protocols

We can now define *interaction protocols*, which constrain the way in which agents can utter messages. Intuitively, an interaction protocol is a ConDec protocol where variables are utterances.

Definition 2. (Interaction protocol) Let a *vocabulary* V be a set of words, and A be a set of agent IDs. $M = A \times V$ is the set of *messages* over V and A . An *interaction protocol* is a ConDec protocol $\mathfrak{P} \subseteq Cons(M)$. ■

To simplify we include only the sender agent in the message, which is useful for interactions between two agents or when messages are always sent to all participants. In Section

5.3 we discuss other situations. The semantics of an interaction protocol is defined over interactions that represent a sequence of uttered messages.

Definition 3. Let M be a set of messages. An *interaction* over M is a finite sequence $I \in M^*$. ■

Any operation over sequences can be applied to an interaction I . We will use the length function ($len(I)$), the function that appends an element ($I.m$), and the notion of prefix (written $I \preceq I'$ if I is a prefix of I'). We will also use indexes ($I(i)$) to obtain the element in the i -th position. From now on, let V , A and M be, respectively, a vocabulary, a set of agent ids, and a set of messages over them, including a message m . Let also \mathfrak{P} and I be, respectively, a protocol and an interaction over M .

Since interaction protocols are essentially LTL theories, we will define their semantics using the existing definitions for that logic. To this end it is only necessary to encode interactions into infinite sequences of states that are labeled with subsets of M . Let I be an interaction over messages M and S be a set of states and $L : S \rightarrow 2^M$ a labeling function.

Definition 4. (Satisfiability) Let w_I be the infinite sequence of states in S such that:

1. $L(w_I(i)) = \{I(i)\}$ for $1 \leq i \leq len(I)$
2. $L(w_I(i)) = \emptyset$ for $i > len(I)$

We call I a *model* of \mathfrak{P} (noted $I \models \mathfrak{P}$) if $w_I \models \mathfrak{P}$. We call I a *partial model* of \mathfrak{P} (noted $I \models_p \mathfrak{P}$) if it is a prefix of a model of \mathfrak{P} , that is, there exists I' such that $I \preceq I'$ (I is a prefix of I') and $I' \models \mathfrak{P}$. Sometimes we will refer to the satisfiability of a single constraint c . In those cases we will use $I \models c$ for $I \models \{c\}$ and $I \models_p c$ for $I \models_p \{c\}$. ■

Definition 4 implies that checking satisfiability of an interaction protocol is equivalent to checking LTL satisfiability, and is therefore decidable.

3.3 Compatibility between Protocols in Different Languages

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 this means. Let us start by defining the notion of vocabulary translation. From now on, let V' be another vocabulary, and $M' = A \times V'$ a set of messages with the same agents as M , but vocabulary V' . Let also $\mathfrak{P}' \subseteq Cons(M')$ be a protocol.

Definition 5. A *translation* between V and V' is a function $\tau : V \rightarrow V'$. τ can be extended homomorphically to a function between:

- messages in $M = A \times V$ and in $M' = A \times V'$ ($\tau : M \rightarrow M'$)
- constraints over M and M' ($\tau : Cons(M) \rightarrow Cons(M')$)
- interactions over M and M' ($\tau : M^* \rightarrow M'^*$) and sets of interactions ($\tau : 2^{M^*} \rightarrow 2^{M'^*}$)
- protocols over M and M' ($\tau : 2^{Cons(M)} \rightarrow 2^{Cons(M')}$) ■

We will call *mapping* a pair of words from different vocabularies. We will say a mapping between words $v \in V$ and $v' \in V'$ is *correct* for a translation τ if $\tau(v) = v'$, or incorrect otherwise.

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 in having the same models modulo a translation.

Definition 6. Let $Int(\mathfrak{P}) = \{I \in M^* \text{ such that } I \models \mathfrak{P}\}$ be the set of models of a protocol \mathfrak{P} . The protocols \mathfrak{P} and \mathfrak{P}' are *compatible* if there exist translations $\tau : V \rightarrow V'$ and $\tau' : V' \rightarrow V$ such that

$$\begin{aligned} Int(\mathfrak{P}) &= \tau(Int(\mathfrak{P}')) \\ Int(\mathfrak{P}') &= \tau'(Int(\mathfrak{P})) \end{aligned}$$

In many cases we will be working with a bijective translation τ such that $\tau^{-1} = \tau'$. If we know the translations τ and τ' for which this condition holds, we can say they are *compatible under τ* . ■

Example (ordering drinks). Let us formalize ConDec protocols for an interaction to order drinks, alternative to the specification in Figure 1. Consider again a waiter W and a customer C interacting to order drinks. Let the vocabulary of the customer be $V_C = \{\text{to drink, beer, wine, water, size, pint, half pint}\}$, and that of the *Waiter* be $V_W = \{\text{da bere, birra, vino, acqua, tipo, media, piccola}\}$. Consider the bijective translation $\tau : V_W \rightarrow V_C$ such that $\tau(\text{da bere}) = \text{to drink}$, $\tau(\text{birra}) = \text{beer}$, $\tau(\text{vino}) = \text{wine}$, $\tau(\text{acqua}) = \text{water}$, $\tau(\text{tipo}) = \text{size}$, $\tau(\text{media}) = \text{pint}$, $\tau(\text{piccola}) = \text{half pint}$.

The following protocols can specify the *ordering drinks* interaction. \mathfrak{P}_C is the protocol used by the customer, with English messages, while \mathfrak{P}_W is the one of the waiter, in Italian. Both agents agree on the first six constraints, which describe what can happen after the waiter asks what the customer wants to drink. For example, the fourth constraint states that if the customer orders beer, then at some point after that the waiter has to ask for the size (or *tipo* in Italian, in this particular interaction). The protocol of the waiter, however, has one extra constraint, stating that beer and wine cannot be ordered together. This is not a constraint for the customer.

$$\begin{aligned} \mathfrak{P}_C &= \{existence(1, \langle W, \text{to drink} \rangle), \\ &\quad premise(\langle W, \text{to drink} \rangle, \langle C, \text{beer} \rangle), \\ &\quad premise(\langle W, \text{to drink} \rangle, \langle C, \text{wine} \rangle), \\ &\quad \quad response(\langle C, \text{beer} \rangle, \langle W, \text{size} \rangle), \\ &\quad premise(\langle W, \text{size} \rangle, \langle C, \text{half pint} \rangle), \\ &\quad premise(\langle W, \text{size} \rangle, \langle C, \text{pint} \rangle)\} \end{aligned}$$

$$\begin{aligned} \mathfrak{P}_W &= \{existence(1, \langle W, \text{da bere} \rangle), \\ &\quad premise(\langle W, \text{da bere} \rangle, \langle C, \text{birra} \rangle), \\ &\quad premise(\langle W, \text{da bere} \rangle, \langle C, \text{vino} \rangle), \\ &\quad \quad response(\langle C, \text{birra} \rangle, \langle W, \text{tipo} \rangle), \\ &\quad premise(\langle W, \text{tipo} \rangle, \langle C, \text{piccola} \rangle), \\ &\quad premise(\langle W, \text{tipo} \rangle, \langle C, \text{media} \rangle), \\ &\quad !correlation(\langle C, \text{birra} \rangle, \langle W, \text{vino} \rangle)\} \end{aligned}$$

The two protocols above are not compatible under any translation. The protocol \mathfrak{P}_C has a model in which the customer orders wine *and* beer, while \mathfrak{P}_W , due to the last constraint, only accepts as models interactions in which only one alcoholic beverage is ordered. If $!correlation(\text{beer}, \text{wine})$ is added to \mathfrak{P}_C , the resulting protocols would be compatible, in particular under τ .

The notion of compatibility will be a key one in our techniques. If agents have compatible protocols, at each state they agree on which messages are possible modulo a translation, provided they interpreted correctly the previous messages. This can be used to obtain information about possible mappings. Concretely, when an agent receives a message it will analyze each possible interpretation to see whether choosing it makes the interaction consistent or not with the constraints, extracting information about the mappings from this. The following defines formally the notion of not being consistent with a constraint.

Definition 7. (Constraint violation) Let $c \in \text{Cons}(M)$ be a constraint. We say c is *violated* by an interaction I if $I \not\models_p c$. If, for interaction I and message m , $I \models_p c$ but $I.m \not\models_p c$, we will say that m violates c in that situation. ■

It is important to note that we say a constraint is violated when the interaction does not *partially* satisfy it. This is because we are interested in the situation for which a message is definitely not possible. For example, we may have $I.\langle a_2, v_1 \rangle \not\models \text{response}(\langle a_2, v_1 \rangle, \langle a_1, w_1 \rangle)$, but this does not mean that a_2 cannot say v_1 , because a_1 could later say w_1 making the interaction satisfy the constraint. A constraint is only violated when no subsequent utterance can make the interaction satisfy it. For some constraints (such as *response*) this is never possible. These constraints can only be identified as not satisfied when observing a complete interaction. Other constraints can be violated by a message before the interaction has finished. The concept of *p-necessity* captures this difference.

Definition 8. Let $c \in \text{Cons}(M)$ be a constraint. We say c is *p-necessary* when, for any interaction I over M , $I \models_p c$. Inversely, c is *non-p-necessary* if there exists an interaction I such that $I \not\models_p c$. ■

We will call *(non)-p-necessities* (non)-p-necessary constraints. In words, a constraint is p-necessary when, even if it is not satisfied by a given interaction, it is partially satisfied by it, i.e, it can always eventually be satisfied by a continuation of the interaction. A constraint is non-p-necessary if there exists an interaction that cannot be continued in any way that satisfies it. In fact, as we will show, this happens for every interaction that is not a model of the constraint. That is, for non-p-necessities, all interactions that are not models are also not partial models.³

The following proposition divides the constraints in our protocols between those that are p-necessary and those that are not.

3. The notions of ‘partial model’, ‘constraint violation’ and ‘p-necessity’ introduced in the previous definitions bear some relationship with related concepts as used in the three-valued semantics for LTL of Bauer et al. (2011) for runtime verification of software systems. In particular, an interaction I is a partial model of a protocol \mathfrak{P} whenever—in LTL_3 — \mathfrak{P} is either true or inconclusive with respect to I ; a constraint c is violated by an interaction I if—in LTL_3 — c is false with respect to I (i.e., I is a “bad prefix”); and a constraint c is p-necessary if no interaction is a “bad prefix” for satisfying c .

Proposition 1. *The constraints existence, correlation, and response are p-necessary. All other constraints are non-p-necessary.*

PROOF. Let $\#(m, I)$ be the number of times that message m appears in I . We will use I^i to denote the suffix of I starting at index i , and $I^{:i}$ for the prefix that finishes at index i . Let us first prove p-necessity. We need to show that $I \models_p c$ for any interaction I . If $I \models c$, then $I \models_p c$ trivially. We consider, for each p-necessary constraint c , an interaction I such that $I \not\models c$, and show that $I \models_p c$.

- For $c = \text{existence}(n, m)$, if $I \not\models c$, then $\#(m, I) < n$. Let I' be an interaction such that $\#(m, I') = n - \#(m, I)$, and consider a new interaction I'' composed of I followed by I' . Then $I \prec I''$, and $\#(m, I'') = n$, so $I'' \models c$, and therefore $I \models_p c$.
- For $c = \text{correlation}(m, m')$, if $I \not\models c$, then either $m \in I$ and $m' \notin I$ or vice versa. Suppose $m \in I$; the other case is analogous. Let $I' = I.m'$. Then $I \prec I'$, and both $m \in I'$ and $m' \in I'$, so $I' \models c$, and therefore $I \models_p c$.
- For $c = \text{response}(m, m')$, if $I \not\models c$, then there exists i , $1 \leq i \leq \text{len}(I)$, such that $I(i) = m$ but $m' \notin I^i$. Taking $I' = I.m'$ as in the previous case also works as proof here.

To prove non-p-necessity, we need to show an example of an interaction I such that, for any I' , if $I \prec I'$ then $I' \not\models c$. As we show, it is enough to consider any I such that $I \not\models c$:

- If $c = \text{before}(m, m')$ and $I \not\models c$, there is an index i , $1 \leq i \leq \text{len}(I)$, such that $I(i) = m'$, but $m \notin I^i$. Then this is also true for any I' if $I \prec I'$.
- If $c = \text{premise}(m, m')$ and $I \not\models c$, then either $I(1) = m'$ or there is an index i , $2 \leq i \leq \text{len}(I)$ such that $I(i) = m'$ and $I(i-1) \neq m$. In any case, this holds for any I' such that $I \prec I'$.
- If $c = !\text{response}(m, m')$ or $c = !\text{before}(m, m')$ and $I \not\models c$, there are indexes i, j ($1 \leq i < j \leq \text{len}(I)$) such that $I_i = m$ and $I_j = m'$. This is also true for any I' if $I \prec I'$.
- If $c = !\text{correlation}(m, m')$ and $I \not\models c$, then $m \in I$ and $m' \in I$. If $I \prec I'$, then also $m \in I'$ and $m' \in I'$.
- If $c = !\text{premise}(m, m')$ or $c = !\text{limm_after}(m, m')$ and $I \not\models c$, there is $I(j) = m$ and $I(j+1) = m'$ for some $1 \leq j < \text{len}(I)$. If $I \prec I'$, this is also true in I' .
- If $c = !\text{existence}(n, m)$ and $I \not\models c$, then $\#(m, I) > n$. This is the same for any I' if $I \prec I'$. \square

Corollary 1. *Let c be a non-p-necessary constraint and I an interaction. If $I \not\models c$, then $I \not\models_p c$.*

PROOF. It is derived from the proof of the proposition, since we did not impose any restriction on the interaction I chosen as an example. \square

4. Communicating with Heterogeneous Partners

Let us now describe the communicative situation that we study in this article. From now on, we consider interactions between two agents a_1 and a_2 , who need to collaborate with each other to perform certain tasks. They both know the tasks to perform, but use different vocabularies V_1 and V_2 respectively. The objective of the methods we propose is to allow agents to learn to understand each other, enabling them to perform the tasks together. Note that to simplify the problem, we consider only interactions between two agents, each of whom speaks only one vocabulary. This does not imply that all interactions need to be with the same partner. As long as the foreign vocabulary remains the same, an agent can change its interlocutor and keep learning. If there is more than one foreign vocabulary, agents need to know which one is used by each interlocutor to apply our method directly. Our method can also be used for interactions between more than two agents. We discuss this simple extension in Section 5.3.

As we mentioned before, the translation technique that we propose is based on the assumption that, for each task, both agents share the procedural knowledge that describes how it must be performed. This can be made more concrete as follows:

1. Both agents agree on a set of tasks that they can perform together.
2. For each task, agents have protocols that are compatible with each other.
3. Agents use their vocabulary consistently throughout different tasks, i.e., the protocols for all tasks are compatible under the same translation.

Formally, let $M_1 = \{a_1, a_2\} \times V_1$ and $M_2 = \{a_1, a_2\} \times V_2$ be sets of messages. Let the function $\tau : V_1 \rightarrow V_2$ be a bijective translation such that, for each task that a_1 and a_2 can perform together, they have protocols \mathfrak{P}_1 and \mathfrak{P}_2 over M_1 and M_2 respectively, and \mathfrak{P}_1 and \mathfrak{P}_2 are compatible under τ . We assume that whenever agents interact, they use protocols that correspond to the same task, or more concretely, compatible protocols.

Note that we assume that the translation τ is bijective. Recall that we defined compatibility as equivalence of all models modulo a translation. If this translation is not bijective, and there are, for example, two words v_1 and v'_1 in V_1 mapped to v_2 in V_2 , these two words would have to be indistinguishable at least in the context of the protocols that are considered. If there is a situation in which, for example, v_1 can be said and v'_1 cannot, the protocols could not be compatible under τ . Therefore, in practical terms, in that context it would be equivalent to consider a vocabulary V'_1 with only one of the words. For this reason, and since it simplifies the exposition significantly, we simply assume that the translation is bijective. This assumption, however, is not necessary for our techniques to work as we report. We will discuss at each point how to handle non-bijective translations.

Finally, we also assume agents have a way of finishing the interaction that their interlocutor is also able to interpret. This can be simply modeled as a shared timeout period after which, if no agent has spoken, the interaction is considered finished. We will later show why this assumption is important.

During an interaction, an agent can send messages using words of its vocabulary and receive others from its interlocutor, or finish the communication if certain conditions hold. We do not force them to follow any particular turn-taking pattern, but we assume that they

both know who speaks at each point. We also assume messages are not lost and arrive on time, at least most of the times. Since we propose a probabilistic approach, we can handle a certain amount of errors, but we do not analyze this aspect in depth here.

Since agents send messages in their own vocabulary, the receiver needs to decode the words it gets, interpreting them in its own terms to be able to continue the execution. Specifically, agents need to learn the translation that allows them to interpret received words correctly, that is, the τ under which their protocols are compatible. We present a general approach to learn τ 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 simplicity, from now on we adopt the perspective of agent a_1 . Since the objective of this agent is to translate words from V_2 , it needs to find the function $\tau : V_2 \rightarrow V_1$.

We propose to learn τ by taking into account which messages are allowed by the protocols at each state. When a_1 receives a message in the vocabulary V_2 used by a_2 , it analyzes which interpretations are allowed by the protocol and which ones are not, and learns from this information. To this end, we use a probabilistic learning technique, in which agents maintain a confidence distribution over possible meanings for foreign words, and update their values with the experience of interacting. Probabilistic techniques are commonly used for all kind of learning problems and particularly for learning word-meaning correspondences (see for example the work by Xu & Tenenbaum 2007), and, as we will explain, they are useful to capture the uncertainty that agents encounter during the learning process. In our problem, an alternative to using probabilistic techniques is possible. It consists on storing relevant information about every observed interaction, and then extracting logical consequences from it. In Section 8 we compare our techniques with one that only uses logical inference.

Definition 9. Let V_1 and V_2 be two vocabularies. An *alignment* is a partial function $\omega : V_1 \times V_2 \rightarrow [0, 1]$. ■

Intuitively, if $v_1 \in V_1$ and $v_2 \in V_2$, the value $\omega(v_1, v_2)$ represents a_1 's confidence in that v_1 is the correct meaning for v_2 , that is, that $\tau(v_2) = v_1$. The function is partial because a_1 does not know V_2 a priori, and therefore it is only defined for those words that the agent received in a message at some point. For the same reason we will keep $\sum_{v \in V_1} \omega(v, v_2) = 1$ but do not require the same in the other direction.

In the following section we will explain in detail how ω is updated with the agent's interacting experiences. Now, let us focus on the dynamics of the interaction; concretely, on how agents send messages, interpret received ones, and finish the interaction.

As we discussed, agents send messages in their own vocabularies that the receivers have to interpret. This implies that the sequence of messages that were actually sent differs from the interaction that each agent interpreted. We will call the former ones *global interactions*, where messages can either be $\langle a_1, v_1 \rangle$ or $\langle a_2, v_2 \rangle$ with $v_1 \in V_1$ and $v_2 \in V_2$. *Local interactions* are the ones interpreted by agents and contain only messages with words in V_1 or only messages with words in V_2 . We will call I_1 and I_2 the local interactions of a_1 and a_2 respectively. To explain the actions of choosing utterances, interpretations, and when to finish the interaction we will make use of the notion of *possible messages*. That is, those messages that can be uttered at a certain point in the interaction.

Definition 10. Let M be a set of messages and \mathfrak{P} a protocol over it. Let I be a local interaction that has happened so far (and $I \models_p \mathfrak{P}$). We call the *possible messages* at that point all $m \in M$ such that when they are added to I , the interaction remains a partial model, that is $I.m \models_p \mathfrak{P}$. ■

Notice, first, that we use the notion of partial satisfiability to define possible messages. That is, a message is possible if, after being uttered, the interaction can be finished in a way that satisfies the protocol. Second, possible messages for a_1 are not necessarily the translation of the ones for a_2 . This is because the local interactions of each agent depend on their interpretations of past messages, which are not necessarily correct.

Now we can explain the dynamics of the interaction.

Choosing messages to send The choice of messages to utter is internal to each agent and depends on its goals while interacting. We do not impose any restriction on this, except for not violating the constraints in the protocol. That is, agent a with protocol \mathfrak{P} can utter a word v after interaction I only if $\langle a, v \rangle$ is a possible message for I and \mathfrak{P} .

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 alignment ω . We assume that, in an interaction, agents choose always the same mapping for a foreign word, and they do not choose the same mapping for two different words. To formalize this we define the function μ , that represents the mappings made in an interaction.

Definition 11. Let I be a global interaction and I_1 agent a_1 's corresponding local interaction. The partial function $\mu : V_2 \rightarrow V_1$ maps all the foreign words received by a_1 in an interaction to the local ones chosen as interpretations. That is, $\mu(v_2) = v_1$ if and only if there exists an index i such that $1 \leq i \leq \text{len}(i)$, $I(i) = \langle a_2, v_2 \rangle$, and $I_1(i) = \langle a_2, v_1 \rangle$. ■

Note that if the interaction has not started ($I = I_1 = []$), then $\text{dom}(\mu) = \emptyset$. If agent a_1 receives v_2 after local interaction I_1 , it will choose a word $v \in V_1$ only if forms a possible message when uttered by a_2 , and if it was not already chosen as an interpretation before. Between the words that satisfy those conditions, it chooses (randomly) one that has highest confidence. Let W be all $v \in V_1$ such that $\langle a_2, v \rangle$ is possible, and such that $v \notin \text{img}(\mu)$. Let *random* be a function that selects a random element from a set. The agent will choose an interpretation for v_2 as follows:

$$\begin{aligned} & \text{random}\left(\underset{v \in W}{\text{argmax}} \omega(v, v_2)\right) && \text{if } v_2 \notin \text{dom}(\mu) \\ & \mu(v_2) && \text{if } v_2 \in \text{dom}(\mu) \text{ and } \langle a_2, \mu(v_2) \rangle \text{ is possible} \end{aligned}$$

In the remaining case, when $v_2 \in \text{dom}(\mu)$ but $\langle a_2, \mu(v_2) \rangle$ is not possible, the agent considers the interpretation failed and it finishes the interaction. Note that the way in which agents choose interpretations makes μ injective.

Finishing the interaction An interaction can finish in two ways. First, an interaction always finishes when an agent has no possible messages to say. 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 (not partial) of the protocol when it ends. In this case, it simply stops talking, producing a time-out that lets the other agent realize the interaction has finished.

5. Learning Translations from Interactions

In the approach we introduce in this paper the alignment ω is updated with the experience that agents collect when interacting. More concretely, agents decide how to update the alignment by analyzing whether an interpretation is possible or not for a received word. Recalling Definition 8, the constraints that can be violated by a new message are those that are not p-necessary. In this article we will focus mainly on the information obtained from this kind of constraints. In Section 7 we discuss how we could also take into account p-necessary constraints, and we discuss why this is a more challenging problem.

The approach that we propose to update ω can incorporate external alignments, as we did in previous work (Chocron & Schorlemmer, 2016) for the case of protocols specified with finite state machines. These alignments represent previous knowledge about the foreign vocabulary, which can have been obtained from previous experience, from a matching tool (Euzenat & Shvaiko, 2013), or with techniques based on semantic vectors (see for example the work by Pennington et al. 2014). The alignments that are obtained with these automatic techniques, however, can never be completely trusted. First, the methods themselves always have a precision lower than 100% for technical reasons (see the work by Cheatham et al. 2015 for a recent evaluation of ontology matching techniques, and the papers they refer to for evaluations of vector-based techniques). In addition, which mappings are considered correct can change from one application context to another, so an alignment with high precision for a context can be less useful for interactions in a different environment. For example, an Italian-English dictionary might translate *piccola* as *small* and *media* as *half* (or *medium*). Therefore, the mapping between *half pint* and *piccola* would hardly if ever be found. In an ordering drinks situation, however, this is the mapping that makes the customer obtain the size of beer she wants.

When external alignments are available, the techniques in this article can be applied as methods for testing and repairing possibly wrong mappings between words. In general, existing tools provide a function that assigns confidence values to mappings, just like our alignments (see Definition 9). An external alignment for agent a_1 is, then, a confidence distribution $\omega_p : V_1 \times V'_2 \rightarrow [0, 1]$, where V'_2 is a set of terms such that $V'_2 \cap V_2 \neq \emptyset$. Even when the alignment ω_p is not defined over V_2 , the information it provides about words in that set can still be used. We interpret the confidences always positively when the confidence is greater than 0, meaning that a mapping with low (but positive) confidence has always more confidence than one that does not exist. If the agent has an alignment that maps words between them instead of assigning values (a translation, as in Definition 5), it can be converted into an alignment by assigning value 1 to each correct mapping.

External alignments are taken into account in the initialization of the values of ω . This is done the first time an agent receives each foreign word. When a_1 receives message $\langle a_2, v_2 \rangle$ for the first time, it assigns a value $\omega(v, v_2)$ for each $v \in V_1$. This is done in a way such

that it takes into account the external alignment alignment, and no word has value 0 or 1, considering in this way that the previous translation could be incorrect. If a_1 has an external alignment ω_p and $v_2 \in \text{dom}(\omega_p)$, it updates ω using that information. For each $v_1 \in V_1$,

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

Then it normalizes the values using an exponential method (we use *softmax*). This ensures that the values for all mappings start in the open interval $(0, 1)$. If there is no external alignment, the agent initializes the interpretation values with a uniform distribution, setting $\omega(v_1, v_2) = \frac{1}{|V_1|}$ for all $v_1 \in V_1$.

After the initialization phase, or whenever a_1 receives a word v_2 that it has already received before, the update phase begins. For each word $v_1 \in V$, the value $\omega(v_1, v_2)$ will be updated according to whether v_1 is or not a possible interpretation. The technique that we propose to this end is inspired in Bayesian updates, combining the previous value of the mapping in ω and the information about which interpretations are possible in the current interaction. Intuitively, if the word v_2 is received after interaction I_1 , we want $\omega(v_1, v_2)$ to express the probability that v_1 is the correct interpretation for v_2 in this situation, that we write $p(\tau(v_2) = v_1 \mid I_1 \cdot \langle a_2, v_2 \rangle)$. According to Bayes' rule this can be expressed as follows:⁴

$$p(\tau(v_2) = v_1 \mid I_1 \cdot \langle a_2, v_2 \rangle) = p(I_1 \cdot \langle a_2, v_2 \rangle \mid \tau(v_2) = v_1) p(\tau(v_2) = v_1) \quad (1)$$

Let us analyze this expression. First, $p(\tau(v_2) = v_1)$ is the prior probability that v_1 is a correct interpretation of v_2 . This information can be obtained from the alignment $\omega(v_1, v_2)$. Since we keep the function normalized for $v \in V_1$, we can use the value directly. Second, the value $p(I_1 \cdot \langle a_2, v_2 \rangle \mid \tau(v_2) = v_1)$ represents the probability that v_2 is said by a_2 after I_1 if the correct translation of v_2 is v_1 . This is equivalent to the probability that v_1 is said after I_1 . Therefore we can rewrite Equation 1 as:

$$p(\tau(v_2) = v_1 \mid I_1 \cdot \langle a_2, v_2 \rangle) = p(I_1 \cdot \langle a_2, v_1 \rangle) \omega(v_1, v_2) \quad (2)$$

Since it only involves words in the local vocabulary, the probability that a_2 says v_1 after I_1 can be computed as follows:

$$p(I_1 \cdot \langle a_2, v_1 \rangle) = \begin{cases} 1 & \text{if } \langle a_2, v_1 \rangle \text{ is possible for } I_1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

These updates are only meaningful if agents are completely sure that they interpreted every foreign message up to I_1 correctly and, moreover, that the same is true for their interlocutor. That is, it is necessary that $\tau(I_2) = I_1$. Not taking this into account can be problematic when a message $\langle a_2, v_1 \rangle$ is not possible. According to Equation 3, the probability that v_2 should be interpreted as v_1 is 0. However, it might be that $\tau(v_2) = v_1$, but that the

4. If τ is bijective, the denominator in Bayes' rule, also known as the *marginal probability* of $I_1 \cdot \langle a_2, v_2 \rangle$, can be omitted directly, since $\sum_{v \in v_1} p(I_1 \cdot \langle a_2, v_2 \rangle \mid \tau(v_2) = v) p(\tau(v_2) = v) = 1$. Otherwise, it should be included as a normalizing constant.

message $\langle a_2, v_1 \rangle$ is not possible because of a previously misinterpreted word, either by a_1 or a_2 . This happens, for example, if \mathfrak{P}_1 has a constraint $!response(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ and a_1 interpreted a word incorrectly as v'_1 before. Since it is not possible to ensure that previous messages were correctly interpreted by everyone, we present a way of taking this uncertainty into account when updating ω . To this aim we introduce the distribution p_{sh} , that represents the confidence that the interaction is shared, that is, that $\tau(I_2) = I_1$. If the interaction is not shared, no inference can be made from the observation, and the probability should be kept as $\omega(v_1, v_2)$. Now the alignment is updated by considering both possibilities, weighted by p_{sh} :

$$p(\tau(v_2) = v_1 \mid I_1.\langle a_2, v_2 \rangle) = p(I_1.\langle a_2, v_1 \rangle) \omega(v_1, v_2) p_{sh}(I_1) + \omega(v_1, v_2) (1 - p_{sh}(I_1))$$

As we already mentioned, p_{sh} is impossible to compute exactly. This is because agents have no access to the alignment of their partners, and therefore cannot assign a probability to the words that they interpreted. In Section 5.1 we show two different ways of choosing p_{sh} . Another particularity of our learning setup is that agents contribute directly into building part of the evidence, since they send the messages that will later make up the interaction. In Section 5.2 we discuss how agent can exploit this to learn faster. The performance of these agents is evaluated in the next section.

5.1 Estimating the Confidence in the Interaction

We now present different ways of estimating p_{sh} , the confidence that the interaction is shared, to take it into account in the updates. The first method is independent of the specific constraints in ConDec protocols, and can be used for any constraint system for which satisfiability can be computed. The second one takes into account the particularities of each constraint to make a more fine-grained update.

5.1.1 SIMPLE STRATEGY

A first straightforward approach consists in setting the value of $p_{sh}(I)$ to a constant for any I . This method only takes into account the already defined notion of *possible messages*, without any reasoning about the semantics of ConDec constraints. Therefore, it can be applied to any kind of logical specification, provided that there is a method to decide if an interaction is a model of a protocol. Concretely, the resulting update function decreases the confidence in a mapping if the resulting message is not possible, and leaves the same value otherwise. This corresponds to the Randomized Weighted Majority (RWM) update proposed by Blum and Mansour (2007) for learning from experience by minimizing regret. Formally, let $r \in [0, 1]$ be a punishment parameter. The update for each $v \in V_1$, when v_2 is received after interaction I , is as follows:

$$\omega(v_1, v_2) := \begin{cases} \omega(v_1, v_2) (1 - r) & \text{if } \langle a_2, v_1 \rangle \text{ is not possible} \\ \omega(v_1, v_2) & \text{otherwise} \end{cases} \quad (\text{RWM})$$

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

The key difference between the Randomized Weighted Majority technique and the Simple Strategy that we present here lies in the way in which a mapping is chosen at each step. In RWM, the agent chooses an action with a probability proportional to its weight. In our Simple Strategy agents use the selection technique we presented in Section 4.

5.1.2 REASONING STRATEGY

The other method we propose is a way of considering the semantics of ConDec constraints in particular, and to use the extra information they contain. To illustrate the importance of this information, recall the *ordering drinks* example, and particularly the waiter’s protocol \mathfrak{P}_W . Consider an interaction in which the customer said **water** but the waiter interpreted it as **vino** (meaning *wine* in Italian). If the customer says **beer**, the waiter may think that interpreting it as **birra** (meaning *beer*) 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. We now present an approach that lets agents make use of the semantics of violated constraints to perform a more fine-grained update of the alignment ω .

If a_1 finds that v_1 is not a possible interpretation for v_2 , consider all constraints $c \in \mathfrak{P}_1$ that are violated by $\langle a_2, v_1 \rangle$. The idea is to identify, for each violated constraint, which are the mappings that can be wrong, thus leading to the violation. The mapping between v_1 and v_2 is always a candidate, but not the only possibility, since previous mappings may also be involved. To take this into account, when an interpretation is not possible the value of its corresponding mapping is updated as indicated in Table 2, according to which constraints were violated. The updates are performed iteratively for each violated constraint. After they are completed, all values are normalized to obtain $\sum_{v \in V_1} \omega(v, v_2) = 1$. In this case, we need to use a normalization method that maintains the values that equal 0.

To find possibly wrong mappings agents use the function μ from Definition 11 that provides information about which mappings were made in an interaction. Concretely, we use μ^{-1} to obtain the foreign term that was mapped with a local word. Note that this is only possible under the assumption of bijectivity of τ . When this is not the case, the confidence with which v_1 was a correct interpretation should be $\prod_{v \in \mu^{-1}(v_1)} \omega(v_1, v)$, taking $\mu^{-1}(v_1)$ as a set, instead of $\omega(v_1, \mu^{-1}(v_1))$.

Let us explain the motivation for each update. Violating the negative existential constraint is different to violating a relation constraint, because it expresses a condition over only one message. Therefore if the chosen interpretation violates it, it must be wrong. For this reason, the value of the mapping is set to 0. Similarly, if *before* depends on a message by an agent that has not said anything the mapping is definitely wrong and set to 0.

When a negative relational constraint is violated, there is one message that might have been misinterpreted. We take this into account by subtracting a value that is proportional to the confidence in the mapping of that message with its interpretation. In the case of *premise*, the constraint depends on the immediately previous message. When the constraint requires that it is sent by an agent different to who actually sent the last message, the mapping being considered is impossible and its value is set to 0. Otherwise, if the message was sent by a_2 , the mapping that may have been misinterpreted is the last one, so the update subtracts a value that is proportional to the mapping of the last received word. In the case

Violated Constraint	Update
$!existence(n, \langle a_2, v_1 \rangle)$	$\omega(v_1, v_2) := 0$
$!correlation(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ $!response(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ $!before(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ $!premise(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ $!imm_after(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$	$\omega(v_1, v_2) := \omega(v_1, v_2) - r \cdot \omega(v'_1, \mu^{-1}(v'_1))$
$premise(\langle a, v'_1 \rangle, \langle a_2, v_1 \rangle)$	$\omega(v_1, v_2) := 0$ if the last message in I_1 was not uttered by a <hr/> $\omega(v_1, v_2) := \omega(v_1, v_2) - r \cdot \omega(v''_1, \mu^{-1}(v''_1))$ if $a = a_2$ and $\langle a_2, v''_1 \rangle$ is the last message in I_1
$imm_after(\langle a_2, v'_1 \rangle, \langle a_2, v''_1 \rangle)$ (if $v'_1 \neq v_1$)	$\omega(v_1, v_2) := \omega(v_1, v_2) - r \cdot \omega(v'_1, \mu^{-1}(v'_1))$
$before(\langle a, v'_1 \rangle, \langle a_2, v_1 \rangle)$	$\omega(v_1, v_2) := 0$ if a did not send any message
Other	$\omega(v_1, v_2) := \omega(v_1, v_2) - r \cdot \omega(v_1, v_2)$

Table 2: Reasoning Strategy: Updates for each violated constraint, where $r \in [0, 1]$ is a punishment parameter.

of *imm_after*, the constraint can be violated if it expected a different word after the last message.

In other cases it is impossible to determine the message that might have been wrongly interpreted. This happens, for example, when *before* is violated (and it does not fall in the case above), when the violated constraint depends on a message that a_1 has sent, or when there is no violated constraint. In these cases agents use the default punishment r .

Lastly, in our experiments we choose $r = \frac{1}{|V_1|}$ because then our technique has the effect of subtracting a larger value when the agent is confident in the correctness of the previous interpretation.

5.2 Students and Teachers: Agents that Actively Want to Learn

The agents we considered in the previous section have as their main objective to complete a task, and learning a translation is a byproduct of this process. However, as we mentioned, the interaction I is not an externally observed part of the environment, but is created by the agents themselves. Interlocutors can therefore decide to utter words that will make the interaction possibly more informative in the future, with the particular objective of learning the translation. In the *ordering drinks* example, this is the difference between an agent that wants to drink wine and will only ask for that, and another one that goes to the bar with

the main objective of learning the vocabulary, and will order anything that will likely lead to an interesting interaction.

In this section we explore some heuristics that can be used by agents to try to actively improve their learning rate. Let us first define a new class of possible messages, called *predecessors*.

Definition 12. A word $v \in V_1$ is *local predecessor* for agent a_1 if there is a relational non-p-necessary constraint in \mathfrak{P}_1 that is not *before* or *premise*, and such that the first message of the constraint is $\langle a_1, v \rangle$, and the second one is $\langle a_2, v' \rangle$ for some $v' \in V_1$. The word v is also a predecessor if $!correlation(\langle v', a_2 \rangle, \langle v, a_1 \rangle) \in \mathfrak{P}_1$ for some $v' \in V_1$

A *foreign predecessor* is defined equivalently, but requiring that the constraint’s second message (or possibly the first one for *!correlation*) is a_1 . ■

An interaction that includes predecessors will more likely be informative in the future than one that does not. This is because agents learn when constraints are violated, and some relational constraints can only be violated when a predecessor was uttered. We propose two different agents based on how they choose to utter predecessors.

- *Students*: These agents choose local predecessors with higher probability. Concretely, they make unsaid local predecessors twice as likely as the rest of possible messages.
- *Teachers*: These agents choose foreign predecessors with higher probability. Concretely, they make unsaid foreign predecessors twice as likely as the rest of possible messages.

While ‘students’ try to utter words that will be more informative for themselves in the future, ‘teachers’ choose those that will be useful for their interlocutors. As we show in the evaluation, ‘teachers’ make a better use of the available information. This is because foreign predecessors are more useful to learn from than local ones. With local predecessors, the agent that utters a word does not know whether the other agent has interpreted it correctly, but it still makes inferences about it. Foreign predecessors, instead, are interpreted by the same agent that will later use the rules. If they are interpreted correctly, the inferences drawn from them will also be correct. If they are interpreted incorrectly, no information is obtained. To illustrate this difference, consider an agent a_1 that says v_1 because it has a constraint $!response(\langle a_1, v_1 \rangle, \langle a_2, v'_1 \rangle)$. After this, it will infer that all the words it receives are not mapped with v'_1 . If a_2 interpreted v_1 wrongly and is violating the constraint, these inferences could be incorrect. If instead the constraint is $!response(\langle a_1, v_1 \rangle, \langle a_1, v'_1 \rangle)$, it will be a_2 who will use the information, but only if it interprets v_1 correctly.

5.3 Interactions between More Than Two Agents

All the techniques that we introduced until now assume a scenario in which two agents interact with each other repeatedly. These restrictions allow us to explain the methods in a simple way, making each agent correspond to a different vocabulary. However, our solutions can be easily extended to scenarios in which more than two agents interact, provided that everyone tracks which vocabulary each agent uses (but not necessarily its meaning, of course). In what follows we explore how these extensions can be implemented.

First, let us consider the case in which there are several agents, but only two vocabularies. The simplest case corresponds to a scenario where interactions are still between two agents, but not necessarily the same ones. For example, consider an English speaking customer that interacts with multiple Italian speaking waiters. In this case the techniques can be applied directly; the fact that the agents change does not have any effect on the learning process. A reasonable question in this scenario is whether an agent can learn if it is constantly facing new interlocutors who do not necessarily learn its own vocabulary. In Section 6.2 we show experimentally that this does not stop the learning process.

A more complex case concerns a situation in which interactions can include more than two agents, each of them using one of two vocabularies. These interactions can be specified in different ways:

1. All messages are sent to everyone. This is useful to specify, for example, a conversation between many participants, where each of them talks to everyone else.
2. Messages have one target in the interaction. This can be specified formally by adding a second agent to messages, representing the receiver. Correspondingly, constraints in protocols will be expressed over this type of messages.
3. Some messages have one target, while others are for everyone else. This can be specified by adding a second agent to messages, and including a special name, *broadcast*, which signals messages for everybody.

In all cases, we assume the interaction is fully observable by all participant agents. This is a key requisite since agents need to know what has been said to learn from it. Our technique can be applied for interactions with many participants in any of the three cases before. Let A_1 be the set of agents that use vocabulary V_1 , and A_2 the set of agents that uses vocabulary V_2 . An agent $a_1 \in A_1$ can use our techniques by treating any message uttered by agents in A_1 as uttered by itself, and any message uttered by agents in A_2 as uttered by the interlocutor. Since agents only reason about interpretations they made, they can consider all foreign words in the same way.

Finally let us consider the case in which there are more than two vocabularies simultaneously. To learn two or more vocabularies in parallel, an agent needs to maintain one alignment for each of them with its own vocabulary. Our techniques can be applied in the same way. This holds even when there are more than two vocabularies in a single interaction. Agents can still use the reasoning strategy, even combining interpretations for words in two different vocabularies. Again, this is possible because all the reasoning is performed on local interpretations.

All these extensions assume that agents know which vocabulary everybody else speaks. It is possible to relax this restriction further by learning first to identify vocabularies. We leave an exploration of this problem for future work.

6. Experimental Evaluation

In this section we present experimental results that show the performance of each agent type in different situations. In the first subsection we compare the performance of different agent

types. We analyze an agent using the Randomized Weighted Majority technique, agents using our simple and reasoning strategies (that we call ‘simple’ and ‘reasoner’ respectively), and the effects of using the student and teacher heuristics. In the second part of this section we explore how agents perform in other scenarios, namely when they have external alignments, when the words in the vocabularies do not appear uniformly in the protocols, and when only one of the two agents is learning. The hypotheses that we intend to test with the experiments are:

1. With repeated interactions, agents improve their knowledge about the correct translation.
2. Agents that use our learning techniques outperform those that use a traditional learning method such as the Randomized Weighted Majority technique.
3. Agents that use the reasoning strategy (Section 5.1.2), as well as the ones those choose the words they send to learn more (Section 5.2), outperform agents following the simple strategy (Section 5.1.1) in terms of the amount of interactions that are necessary to discover the correct translation.

Before presenting the results, let us briefly discuss the generation of the data we used and explain the experimental setup.

DATA GENERATION

The protocols that we use in this article have an abstract representation, making it possible to adapt the techniques that we propose to many different more concrete formalizations. However, adapting them to available real-world examples is not trivial, and the generality can be lost in some cases. In this article we evaluate the techniques that we propose with randomly generated data, which allows us to abstract away from any implementation details.

The procedure to generate a protocol starts from a vocabulary, and consists in choosing constraints randomly and adding them to the protocol if it remains satisfiable with the new constraint. Since we evaluate techniques that use non-p-necessary constraints, we included only this type of rules. We generated only *!existence* constraints with $n \leq 2$. We created protocols with different sizes in vocabulary and in the set of constraints. In previous experiments we had simplified the constraints used in the protocols, considering them to be expressed over only words, instead of tuples of words and agents, which allowed us to experiment with larger protocols in less running time (Chocron & Schorlemmer, 2017b). For the experiments reported in this article we consider the version that we defined in 3.1, where a constraint is expressed over a tuple $\langle agent, word \rangle$. We used the NuSMV model checker (Cimatti et al., 2002) to perform all the necessary satisfiability checks.

Starting from one protocol \mathfrak{P}_1 over vocabulary V_1 , a compatible protocol \mathfrak{P}_2 can be generated in a simple way by taking $\tau(\mathfrak{P}_1)$, where $\tau : V_1 \rightarrow V_2$ is a bijective alignment created for a different vocabulary V_2 . Note that this implies that all our evaluation is done for bijective translations.

EXPERIMENTAL SETUP

The objective of our experiments is to determine how fast agents can learn a correct translation between their vocabulary and the one of their interlocutor. A run of an experiment consists of two agents a_1 and a_2 with vocabularies V_1 and V_2 who are sequentially given pairs of protocols compatible under one same translation τ . We implicitly assume that, after many tasks have been executed, there is only one translation between V_1 and V_2 under which all the pairs of protocols are compatible. Agents interact following each pair of protocols, with the dynamics presented in Section 4. To simplify the termination of the interaction, we used bounds that all agents agreed upon to determine the maximum length of an interaction; however, this does not have any effect on the learning process.

We let agents interact 300 times. After each interaction, we measured how close agents were to the correct translation τ . To this aim, we first need to define a measure of how well an agent knows a translation. Since agents always choose the possible mapping with highest weight, we can easily extract from the alignment ω a translation that contains, for each foreign word, the interpretation that would be chosen as a first option in the following interaction. We will refer to the translation extracted from the alignment of agent a_1 as τ_1 . The domain of τ_1 are all $v_2 \in V_2$ such that $(v, v_2) \in \text{dom}(\omega)$ for all $v \in V_1$ and $\tau_1(v_2) = \text{random}(\text{argmax}_{v \in V_1} \omega(v, v_2))$. Note that if there are multiple possibilities with the same value, τ_1 randomly chooses one of them.

To compare the local translations τ_1 and τ_2 with the reference τ , we used the standard precision and recall measures, and their harmonic combination, commonly known as *F-score*. Given two alignments β and γ expressed as relations, the *precision* of β with respect to γ is the proportion of the mappings in β that are also in γ :

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

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

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

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

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

In our case, precision and recall can be different in the first interactions, when agents may still have not observed the complete foreign vocabulary. After the interlocutor has uttered all the words in the vocabulary at least once, precision and recall are equivalent.

We performed experiments parametrized with a protocol and vocabulary size. We initially performed the experiments with the same vocabulary sizes that are used for testing by Atencia and Schorlemmer (2012), 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 agent type. 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 12 words and four different protocol sizes. We used the values $r = 0.3$ for the punishment parameter of the simple strategy.

Each experiment was repeated 10 times, and we averaged the results. The *F-scores* that we refer to in the figures are the average of the *F-score* of each agent that participates in the interaction.

6.1 Comparing Strategies

The first part of this experimental section compares the performance of different types of agents. To this aim, we measured how many interactions agents have to perform before they find the correct alignment. We compared the performance of seven agent types:

- an agent using the Randomized Weighted Majority method, that we call ‘RWM’
- agents ‘simple’ and ‘reasoner’
- agents combining the ‘simple’ and ‘reasoner’ with students and teachers, that we call ‘student simple’, ‘student reasoner’, ‘teacher simple’ and ‘teacher reasoner’.

In all cases we consider pairs of agents of the same type. The plots in Figure 2 show how many of the 100 experiments that were performed achieved an F-score of 1.0 after n interactions. We show the results for agents using protocols with 6, 8, 10 and 12 constraints. The difference is more pronounced for smaller protocols; when there is less information, it is more evident who makes a better use of it. The agent using the existing Randomized Weighted Majority technique is always significantly worse than the ones using our technique, because its way of selecting interpretations is not adequate for the problem. ‘Reasoner’ does better than ‘simple’ because it is able to use more efficiently the information it obtains after some interactions.

Agents always improve their performance when combined with ‘teacher’, because this technique uses the information in the protocols more efficiently, extracting what is more useful. However, as we have already discussed in Section 5.2, ‘student’ is not consistently better than ‘simple’ or ‘reasoner’. This is because they learn from local predecessors very often, which depend on how their interlocutor has interpreted messages. However, they have no information about how these words were interpreted. Given that there is no improvement, from now on we will not consider ‘student’, using only ‘teacher’. Since observing how long agents take to converge shows more clearly the difference between agent types, from now on we will show the results in this way.

6.2 Special Scenarios

Let us now focus on some particular cases that we consider particularly interesting.

6.2.1 ALIGNMENT REPAIR

As we mentioned, the interaction-based alignment technique can be combined in a simple way with external alignments. Of course, these alignments may have incorrect mappings, in which case our learning method works as a repair technique. We studied the performance of our methods when used by agents with external alignments of different qualities, by

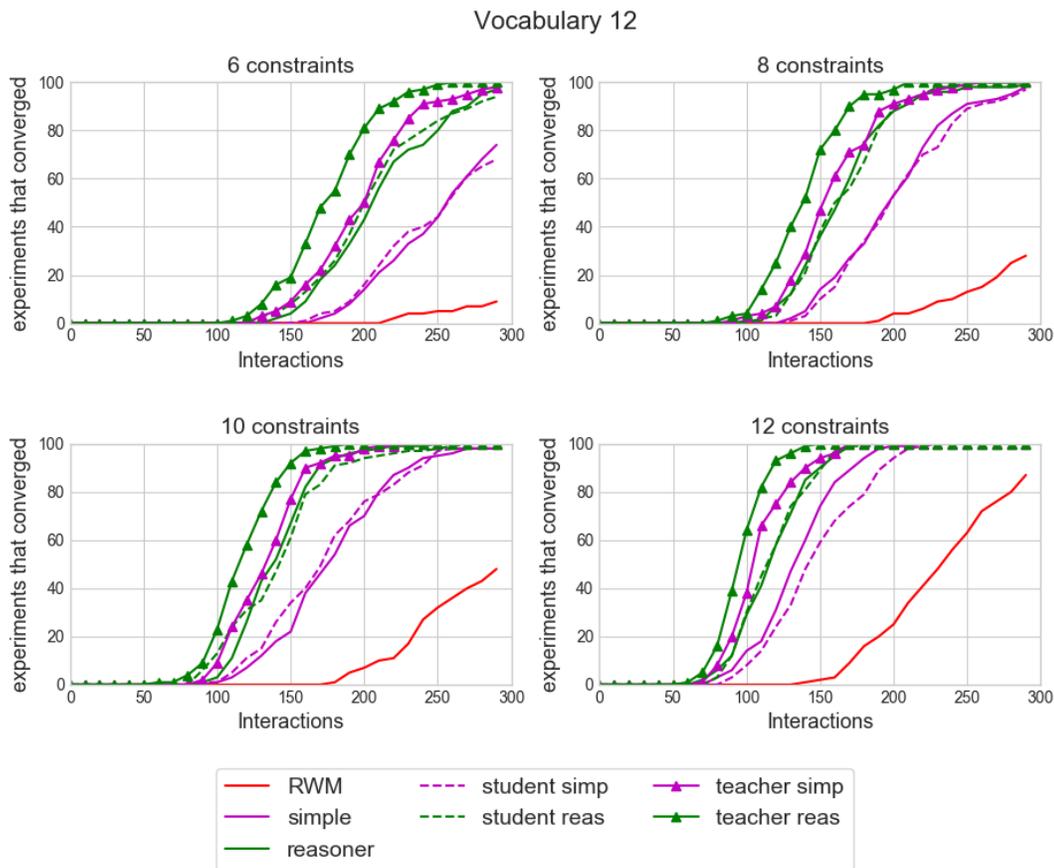
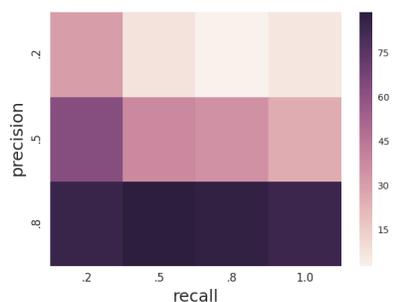


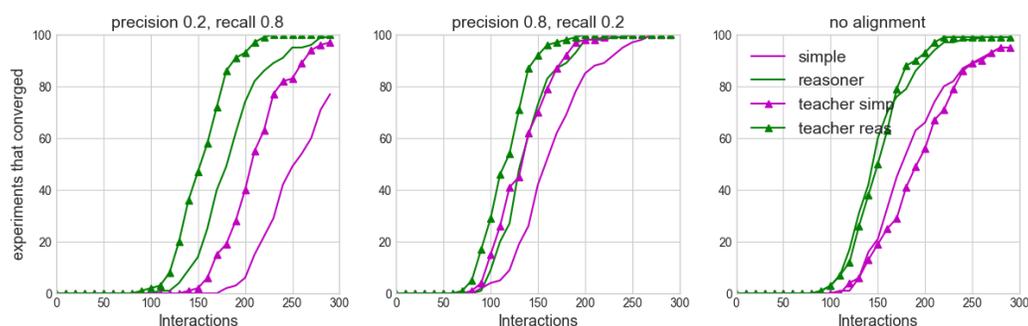
Figure 2: Performance for different types of agents, for protocols of different sizes (measured in number of rules)

varying their precision and the recall with respect to τ . We analyzed the combinations between precision and recall levels of 0.2, 0.5, and 0.8. We also included the recall level of 1.0, to analyze the case in which all correct mappings are found. Again, we performed 100 experiments, with 10 different sets of protocols and 10 repetitions for each set. We used protocols with 8 constraints, and a vocabulary of 12 words.

Figure 3a shows convergence results for ‘simple’ with different values of precision and recall. Since there are some cases that do not converge in 300 interactions, it is impossible to measure the average amount of interactions that are necessary to converge. Instead, we show how many interactions converged in less than 200 interactions. The figure shows an interesting tendency. While the number of interactions that converge grows with the precision of the existing alignment, the same is not observed for the recall. In fact, agents seem to be worse with larger values of recall. This is because a larger recall value for the same precision means that there are more incorrect mappings. According to the results, incorrect mappings are very harmful for the learning process, and agents perform better when they have less of both correct and incorrect ones.



(a) Number of experiments that converged before 200 interactions for different external alignments ('simple' agent).



(b) Results for different agents, with external alignments of two different qualities.

Figure 3: (Alignment repair) Results for agents with previous alignments. Vocabulary of 12 words, protocols of 8 constraints.

Another interesting question is how the different agents perform for each alignment quality. Figure 3b presents the results for the cases when recall is low and precision is high and vice-versa. We include, as a reference, a plot showing the convergence for agents without an alignment. Interestingly, we observe that 'reasoner' improves significantly the performance when there is low precision and high recall. This may be because it has a way of discarding alignments completely, which is useful when there are many of them. This shows that, in addition to learning faster from scratch, our 'reasoner' strategy is better at repairing wrong alignments.

6.2.2 NON-UNIFORM WORD DISTRIBUTIONS IN THE PROTOCOLS

Until now, we have always considered that all words are equally likely to be uttered and to appear in the constraints of a protocol. This is not always the case, and words that appear less frequently can be more difficult to learn. To study how this influences our learning technique, we generated protocols in which some words appeared with less likelihood than others. Concretely, we set a frequency for each word v . When building the protocols, each word was chosen to generate constraints with a probability proportional to its frequency. If

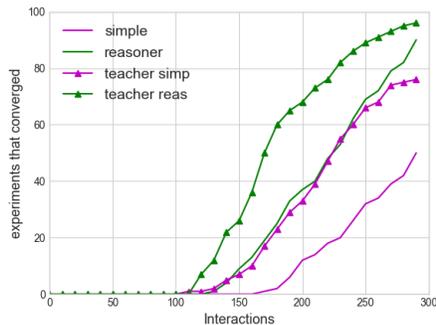


Figure 4: (Non-Uniform Word Distributions) Results for different types of agents, for the case when $\frac{1}{4}$ of the words are half as frequent as the rest.

a word does not appear very often in constraints agents can obtain less information about it, which makes it difficult to learn its translation.

Figure 4 shows the convergence results for the case of 12 words and a protocol with 8 constraints, where three of the words were half as likely as the other ones to be chosen. Since it is more difficult to find that mapping, agents have a slower convergence when compared to the general case in Figure 2. Note that in this case, ‘simple’ is only able to reach convergence in 300 interactions in half of the cases. It is likely that this is because it is very difficult to obtain information about the less common words. ‘Reasoner’ and ‘teacher’, which use information more efficiently, are much better in this case.

6.2.3 UNILATERAL LEARNING

Having two agents trying to learn each other’s vocabulary is not a common situation. While it can be useful in the kind of artificial interactions that we presented, in a more realistic scenario there often exists an established vocabulary in a community, and it is the responsibility of the agent that does not speak it to try to understand it. The other agent may try to understand the messages it receives, but without making an effort to learn. We tested our methods in this situation by implementing a lazy agent that always chooses a random interpretation between the set of possible ones when it receives a message.

In Figure 5 we show the learning process for three types of agents, with vocabulary of 12 words and protocols of size 8. We show results for ‘simple’ and ‘reasoner’. ‘Teacher’ will not have any effect here, since this agent tries to make its partner learn better, but here only one agent learns. The F-score here is the one corresponding to the learning agent, since the lazy agent does not have an alignment. Remarkably, agents (in particular ‘reasoner’) are still able to learn a translation, even when their interlocutor is not getting any better at interpreting their utterances.

6.3 Discussion

The experiments described above show, first of all, that the techniques introduced in previous sections allow agents to learn alignments with foreign vocabularies using only the experience of interacting. We compared our techniques with a well-known learning method based on

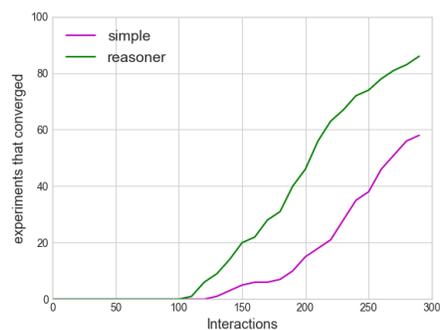


Figure 5: (Unilateral learning) Results for different types of agents, for pairs where only one agent learns.

regret minimization (Randomized Weighted Majority). This method is similar to our simple strategy, but it chooses interpretations in a different way. The experimental results show that our approach works notably better for this problem.

Comparison with other vocabulary alignment techniques is not straightforward because of the type of knowledge used in each case. Most existing techniques use some kind of linguistic or ontological knowledge and assume access to information about the two vocabularies at the same time (Euzenat & Shvaiko, 2013). Some approaches designed for online alignment in multiagent interactions relax the latter assumption (van Diggelen et al., 2006), but still rely on the exchange of ontological information. Our approach, instead, uses only information about procedures. These differences makes it difficult to establish an evaluation framework to compare the techniques, since it is not possible to give all of them the same information. However, we can observe that in most cases our agents achieve, after less than 200 interactions and with randomly generated protocols, a perfect alignment with F-score 1.0. Vocabulary alignment techniques based, for example, on ontological knowledge do not normally achieve these results (Achichi et al., 2016), and the obtained precision depends completely on the amount of information existent for the vocabularies. Our approach is therefore more general and can be applied to the specific vocabulary that the agent wants to align, regardless of the existing resources.

We also showed how our techniques can be useful to repair non-perfect alignments, when words are not equally distributed in constraints, and when only one of the agents is learning. This last experiment is particularly relevant when considering the application of our techniques in situations when, for example, an agent arrives to a new community and has to learn a vocabulary from interacting with different interlocutors. Since it is unrealistic to expect all the community to learn the foreign vocabulary, it is important to have a technique that works when only one side learns.

7. Including p-Necessary Constraints

Until now, we have presented techniques that exploit the information that can be obtained from constraints that are not p-necessary. As we explained, an agent can learn from non-p-necessary constraints when considering possible interpretations for a received word, by

analyzing which candidate interpretations violate a constraint. Considering only the violation of non-p-necessary constraints is simple because, for many of them, there are only two candidates for the words that can have been interpreted incorrectly. This allows agents to use specific techniques, as does ‘reasoner’ in Section 5.1.2.

Constraints that are p-necessary, instead, are not affected by new messages; they can only be considered violated when the interaction is over. For example, if $existence(1, \langle a_1, buy \rangle)$ belongs to a protocol, there is always time for a_1 to say *buy* eventually; only when observing a complete interaction one can conclude that the message was never sent.

Even without considering them explicitly, p-necessary constraints provide extra information. A protocol that consists of a mix between p-necessary and non-p-necessary constraints might not be satisfiable even without any violated constraint. Consider, for example, a protocol containing the following set of constraints:

$$\{response(\langle a_1, u \rangle, \langle a_2, v \rangle), !correlation(\langle a_2, v \rangle, \langle a_1, w \rangle), existence(1, \langle a_1, w \rangle)\}$$

In this case, the message $\langle a_1, u \rangle$ cannot be uttered, since then a_2 would have to say v , making it impossible for a_1 to say w .

Protocols that include p-necessary constraints already take them into account in the simple update we proposed in the first section, since the unsatisfiability of subsets of constraints is considered in the definition of *possible messages*. For example, according to our definition $\langle a_1, u \rangle$ is never a possible message for the protocol above. This effect is more pronounced if the protocols of both agents have extra information specifying that an interaction should finish after a certain number of utterances. We call these protocols *bounded*.

Definition 13. Let M be a set of messages. A bounded protocol over M is a tuple $\langle \mathfrak{P}, n \rangle$ where $\mathfrak{P} \subseteq Cons(M)$, and n is a positive natural number.

The models of a bounded protocol $\langle \mathfrak{P}, n \rangle$ are interactions I over M such that $len(I) \leq n$ and $I \models \mathfrak{P}$. We require that there exists at least one such interaction. Two protocols $\langle \mathfrak{P}_1, n_1 \rangle, \langle \mathfrak{P}_2, n_2 \rangle$ are compatible if \mathfrak{P}_1 and \mathfrak{P}_2' are compatible, and $n_1 = n_2$. ■

For bounded protocols, the definition of possible messages, which remains the same, requires that the interaction can be completed satisfying the protocol in a maximum number of steps, which is more informative.

In Table 3 we show, for different types of protocols, the average of how many interactions agents need to perform before converging to the correct translation. We use a vocabulary of 8 words and different protocol sizes. We compared the case of non-bounded protocols only with constraints that are not p-necessary (such as the ones in the previous section), and of both bounded and non-bounded protocols with a mix of p-necessary and non-p-necessary constraints. We also included an agent that has the same mixed protocols, but only takes into account their non-p-necessary constraints, to show that they do play a role in inferring the alignment. In all cases we used the ‘simple’ agent.

Taking into account p-necessary constraints explicitly is more difficult. We explored two possibilities. The first one consists in analyzing the violated p-necessary constraints at the end of an interaction. This can be done in different ways; either punishing the chosen mappings or rewarding those that would satisfy the violated constraints. However, both methods deal with highly uncertain information. For example, consider a simple case in

	Non-p-necessities	Mixed	Mixed, agents using only non-p-necessities	Mixed, bounded
Prot 6	117.72	180.88	215.82	96.68
Prot 8	87.92	116.64	149.26	71.92
Prot 10	69.72	82.78	125.9	61.1

Table 3: Average number of interactions before convergence, for the ‘simple’ agent, with vocabularies of 8 words.

which a_1 finds that $existence(1, \langle a_2, v \rangle)$ was violated in an interaction. First, a_1 cannot know if the interaction finished successfully for a_2 , or if it finished because there was no possible message. Second, the available information only allows a_1 to either punish all mappings made or to reward the mapping of each received word with v . Since only one of those words needs to be v , this implies all the rest are wrong updates. The second approach does not use violated p-necessary constraints, but those that were satisfied. It consists on rewarding those mappings that made a p-necessary constraint become valid. In the case before, if w was mapped with v , the agent would reward this mapping. However, this also resulted in rewarding many wrong mappings.

Due to the high uncertainty, none of these approaches resulted in significant improvements in the performance. In some cases they were even worse than the basic case. It remains as future work to develop useful techniques for p-necessary constraints, or to analyze the case in which protocols consist of only constraints of this kind.

8. A Purely Logical Approach to Alignment Inference from Interactions

From a technical perspective, the work we presented in the past sections is a way of using learning techniques to extract information from logical specifications. There exist other inference methods to obtain this kind of information, some of which have been extensively studied, that are not based on probabilistic techniques. Instead, these methods use the logical information directly to make deductions. In this section we introduce an ad-hoc logical agent that extracts all the available information from the interactions it is involved in. This method extracts more precise information from the experience, since it can take into account the exact information about the relations that it sees. However, it has no way of extracting information from an uncertain situation, which can only be considered with a probabilistic approach. For example, it can learn nothing at all when a violated constraint involves a word interpreted by its interlocutor.

We will call ‘logical’ the agent that uses only logical inferences, that works as follow. First, it generates all possible translations. Since it does not know the foreign vocabulary, it represents possible translations as all permutations of its own vocabulary, which will be mapped with a unique sequence of foreign words as soon as the agent learns them. At the beginning of each interaction it chooses a translation between the possible ones to translate received words. Then, it starts interacting with the other agent. When it is its turn to speak, ‘logical’ chooses the utterance in the same way as ‘simple’. Interpretations are chosen according to the selected translation. When a message is received, it analyzes the possible translations and discards those that are impossible according to this new evidence. This is

Violated Constraint	Update
$!existence(n, \langle a_2, v_1 \rangle)$	discard translations τ' if $\tau'(v_2) = v_1$
$!correlation(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ $!response(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ $!before(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ $!premise(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$ $!imm_after(\langle a_2, v'_1 \rangle, \langle a_2, v_1 \rangle)$	discard translations τ' if $\tau'(v_2) = v_1$ and $\tau(\mu^{-1}(v'_1)) = v'_1$
$premise(\langle a, v'_1 \rangle, \langle a_2, v_1 \rangle)$	discard translations τ' if $\tau'(v_2) = v_1$ if the last message in I_1 was not uttered by a <hr/> discard translations τ' if $\tau'(v_2) = v_1$ and $\tau(\mu^{-1}(v'_1)) = v'_1$ if $\langle a_2, v''_1 \rangle$ is the last message in I_1
$imm_after(\langle a_2, v'_1 \rangle, \langle a_2, v''_1 \rangle)$	discard translations τ' if $\tau'(v_2) = v_1$ and $\tau(\mu^{-1}(v'_1)) = v'_1$
$before(\langle a, v'_1 \rangle, \langle a_2, v_1 \rangle)$	discard translations τ' if $\tau'(v_2) = v_1$ if a did not send any message
Other	nothing

Table 4: Updates for each violated constraint.

done in a similar way to how ‘reasoner’ updates its alignment, since it is based on the same information. Table 4 shows the actions for each case. If the chosen translation is updated as impossible, agents finish the interaction.

To evaluate the performance of ‘logical’ we performed the same kind of experiments as before, letting agents interact for 500 interactions. We used a vocabulary of 8 words and protocols with both p-necessary and non-p-necessary constraints. As before, we measure how many interactions agents perform until they both find the correct translation, that is, when their set of possible translations has only one element. We compared this value to the number of interactions after which ‘simple’ and ‘reasoner’ obtain an F-score of 1. As before, we averaged the values for 100 experiments, performing 10 repetitions for 10 different protocol sets.

Table 5 shows the convergence results for different agents and protocol sizes. ‘Logical’ takes more to converge than ‘simple’, even when the latter one uses no specific information about the type of constraints. When comparing ‘logical’ to ‘reasoner’, which uses the same information, we can observe that ‘reasoner’ is significantly faster.

The time of convergence is, however, not the main problem of ‘logical’. The vocabularies of 8 words that we used in the experiments were the larger ones for which the technique could be executed on our server. After that, it became prohibitively space-consuming, and

	‘simple’	‘reasoner’	‘logical’
6 constraints	215.82	164.14	310.82
8 constraints	149.26	112.4	203.5
10 constraints	125.9	96.1	157.84

Table 5: Convergence for vocabularies of size 8

was automatically killed. In our learning techniques, the number of possible mappings in ω grows polynomially with the size of the vocabulary (n^2). The number of possible bijective translations grows much faster, proportionally to the factorial ($n!$). In addition, while the mappings are two numbers, the translations are n mappings in themselves. This makes the logical technique use much more memory, and also much more computing time since the techniques require to perform searches through all the space. The logical agent can be optimized to use less space, but learning agents do it automatically and converge faster.

In addition, learning agents achieve partially correct translations while learning, which means that they can start interacting successfully earlier, even when they still have not discovered the complete translation.

Another important advantage of probabilistic learning techniques is that they can incorporate possibly erroneous previous data. As we discussed in earlier sections, our methods allow to consider external alignments even when not all their mappings are correct, and are able to repair those wrong correspondences. A logical approach, on the other side, has no way of considering possibly wrong mappings. For similar reasons, the agents that we described in previous sections are able to deal with some errors in the transmission. To this end, it is only necessary that agents do not assign 0 to the confidence on any mapping, or that they normalize using a method that leaves all the values in the open interval $(0, 1)$. ‘Logical’, instead, cannot handle this possible erroneous evidence in a simple way.

Finally, ‘logical’ cannot deal with non-bijective translations as easily as the agents that use our techniques. Consider a situation in which an interpretation violates, for example, a constraint $!response(v_1, v'_1)$, and the agent has mapped both v_2 and v'_2 with v_1 . The logical agent does not know which of these mappings was incorrect, if any, and therefore it cannot learn anything from this situation. As we mentioned before, ‘reasoner’ can easily incorporate this uncertainty by considering the product of the confidences of both mappings.

9. 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. Importantly, we only assume agents to share the knowledge of how to perform tasks, which is necessary to interact even for partners who use the same vocabulary. Our technique can be applied by agents autonomously, without requiring prior coordination with their interlocutors. 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 by integrating our techniques with external alignments. With our methods, agents learn dynamically while interacting and they rapidly achieve a reasonable level of knowledge about the alignment, while the last

mappings are difficult to find. Therefore, a possible way of using our techniques would be to go first through a training phase in which agents learn many mappings, and then to start performing real interactions. While interacting, agents could keep discovering the remaining mappings. We required 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 obtaining a slower learning process, since an extra component of uncertainty is added.

This research opens many venues for future exploration. 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 it possible to reason about p-necessary constraints when an interaction ends. An approach like this would relate our work to the one by Santos et al. (2016) about dialogues for meaning negotiation.

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. In this direction, it is also important to investigate ways of scaling the methods. When using very large vocabularies, it would be prohibitively expensive to maintain a dictionary of confidence values for mappings between all words. A possible approach would be to develop pruning techniques that remove mappings with low confidence. Another method could focus on learning first mappings for a subset of common words, using them later to infer the meaning of the rest. These ideas need to be investigated before our methods can be applied in a real scenario.

Finally, it would be interesting to investigate how the assumption of protocol compatibility can be relaxed. Since our technique uses a probabilistic update, it admits non-compatible protocols if they are not frequent, considering them as noise in the learning process. However, a more explicit investigation of how differences between protocols affect the method needs to be performed. One possibility is the one in our previous work (Chocron & Schorlemmer, 2017a), where we propose a new type of *flexible* protocol where each constraint has a weight. In that situation, a good translation is one that makes agents pay less while interacting. In future work, we plan to further develop this direction.

Acknowledgments: This research has been funded by the European Union’s Seventh Framework Programme under grant agreement no. 607062 *ESSENCE: Evolution of Shared Semantics in Computational Environments*, Horizon 2020 FET Proactive project *WeNet — The Internet of Us*, under grant agreement no. 823783, and by CSIC’s Proyectos Intramurales Especiales *DIVERSIS* (no. 201750E064) and *SMA* (no. 201550E040). We thank the anonymous reviewers whose comments and suggestions helped improve and clarify this manuscript.

References

- KARL ABERER, PHILIPPE CUDRÉ-MAUROUX, & MANFRED HAUSWIRTH. The chatty web: emergent semantics through gossiping. In *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pages 197–206, 2003. doi: 10.1145/775152.775180.
- MANEL ACHICHI, MICHELLE CHEATHAM, ZLATAN DRAGISIC, JÉRÔME EUZENAT, DANIEL FARIA, ALFIO FERRARA, GIORGOS FLOURIS, IRINI FUNDULAKI, IAN HARROW, VALENTINA IVANOVA, ERNESTO JIMÉNEZ-RUIZ, ELENA KUSS, PATRICK LAMBRIX, HENRIK LEOPOLD, HUANYU LI, CHRISTIAN MEILICKE, STEFANO MONTANELLI, CATTIA PESQUITA, TZANINA SAVETA, PAVEL SHVAIKO, ANDREA SPLENDIANI, HEINER STUCKENSCHMIDT, KONSTANTIN TODOROV, CÁSSIA TROJAHN DOS SANTOS, & ONDREJ ZAMAZAL. Results of the ontology alignment evaluation initiative 2016. In *OM: Ontology matching*, pages 73–129, 2016.
- NAMEERA AKHTAR & LISA MONTAGUE. Early lexical acquisition: the role of cross-situational learning. *First Language*, 19(57):347–358, 1999. doi: 10.1177/014272379901905703.
- DAVIDE ANCONA, ANGELO FERRANDO, & VIVIANA MASCARDI. Agents interoperability via conformance modulo mapping. In Massimo Cossentino, Luca Sabatucci, & Valeria Seidita, editors, *Proceedings of the 19th Workshop "From Objects to Agents", Palermo, Italy, June 28-29, 2018*, volume 2215 of *CEUR Workshop Proceedings*, pages 109–115. CEUR-WS.org, 2018.
- DANA ANGLUIN & LEONOR BECERRA-BONACHE. A model of language learning with semantics and meaning-preserving corrections. *Artificial Intelligence*, 242:23–51, 2017. doi: 10.1016/j.artint.2016.10.002.
- MANUEL ATENCIA & MARCO SCHORLEMMER. Semantic alignment of agent interactions through the communication product. In *Proceedings of the 5th European Workshop on Multi-Agent Systems (EUMAS'07), 13-14 December 2007, Hammamet, Tunisia*, pages 416–431, 2007.
- MANUEL ATENCIA & W. MARCO SCHORLEMMER. An interaction-based approach to semantic alignment. *Journal of Web Semantics*, 12:131–147, 2012. doi: 10.1016/j.websem.2011.12.001.
- SIDNEY BAILIN & WALT TRUSZKOWSKI. Ontology negotiation between intelligent information agents. *The Knowledge Engineering Review*, 17(1):7–19, 2002.
- MATTEO BALDONI, CRISTINA BAROGLIO, & ELISA 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.
- MATTEO BALDONI, CRISTINA BAROGLIO, ELISA MARENGO, & VIVIANA PATTI. Constitutive and regulative specifications of commitment protocols: A decoupled approach. *ACM TIST*, 4(2):22:1–22:25, 2013. doi: 10.1145/2438653.2438657.

- SAMUEL BARRETT, NOA AGMON, NOAM HAZON, SARIT KRAUS, & PETER STONE. Communicating with unknown teammates. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 45–50, 2014. doi: 10.3233/978-1-61499-419-0-45.
- ANDREAS BAUER, MARTIN LEUCKER, & CHRISTIAN SCHALLHART. Runtime verification for LTL and TLTL. *ACM Transactions on Software Engineering and Methodology*, 20(4): Article 14, 2011.
- PAOLO BESANA & DAVE ROBERTSON. How service choreography statistics reduce the ontology mapping problem. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, volume 4825 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2007.
- AVRIM BLUM & YISHAY MANSOUR. Learning, regret minimization, and equilibria. *Algorithmic Game Theory*, 01 2007. doi: 10.1017/CBO9780511800481.006.
- MARICELA BRAVO & JOSÉ VELAZQUEZ. Discovering pragmatic similarity relations between agent interaction protocols. In Robert Meersman, Zahir Tari, & Pilar Herrero, editors, *On the Move to Meaningful Internet Systems: OTM 2008 Workshops. OTM Confederated International Workshops and Posters, ADI, AWeSoMe, COMBEK, EI2N, IWSSA, MONET, OnToContent + QSI, ORM, PerSys, RDDS, SEMELS, and SWWS 2008, Monterrey, Mexico, November 9-14, 2008. Proceedings*, volume 5333, pages 128–137. Springer, 2008.
- MICHELLE CHEATHAM, ZLATAN DRAGISIC, JÉRÔME EUZENAT, DANIEL FARIA, ALFIO FERRARA, GIORGOS FLOURIS, IRINI FUNDULAKI, ROGER GRANADA, VALENTINA IVANOVA, ERNESTO JIMÉNEZ-RUIZ, PATRICK LAMBRIX, STEFANO MONTANELLI, CATTIA PESQUITA, TZANINA SAVETA, PAVEL SHVAIKO, ALESSANDRO SOLIMANDO, CÁSSIA TROJAHN DOS SANTOS, & ONDREJ ZAMAZAL. Results of the ontology alignment evaluation initiative 2015. In *Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015*, pages 60–115, 2015.
- PAULA CHOCRON & MARCO SCHORLEMMER. Attuning ontology alignments to semantically heterogeneous multi-agent interactions. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 871–879, 2016. doi: 10.3233/978-1-61499-672-9-871.
- PAULA CHOCRON & MARCO SCHORLEMMER. Vocabulary alignment for agents with flexible protocols. In *Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology, Bozen-Bolzano, Italy, September 21-23, 2017.*, 2017a.
- PAULA CHOCRON & MARCO SCHORLEMMER. Vocabulary alignment in openly specified interactions. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 1064–1072, 2017b.

- ALESSANDRO CIMATTI, EDMUND M. CLARKE, ENRICO GIUNCHIGLIA, FAUSTO GIUNCHIGLIA, MARCO PISTORE, MARCO ROVERI, ROBERTO SEBASTIANI, & ARMANDO TACCHELLA. Nusmv 2: An opensource tool for symbolic model checking. In *Computer Aided Verification, 14th International Conference, CAV 2002, Copenhagen, Denmark, July 27-31, 2002, Proceedings*, pages 359–364, 2002.
- COLIN DE LA HIGUERA. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, New York, NY, USA, 2010. ISBN 0521763169, 9780521763165.
- ULLE ENDRISS. Temporal logics for representing agent communication protocols. In Frank Dignum, Rogier M. van Eijk, & Roberto A. Flores, editors, *Agent Communication II, International Workshops on Agent Communication, AC 2005 and AC 2006, Utrecht, Netherlands, July 25, 2005 and Hakodate, Japan, May 9, 2006, Selected and Revised Papers*, volume 3859 of *Lecture Notes in Artificial Intelligence*, pages 15–29. Springer, 2006.
- JÉRÔME EUZENAT. First experiments in cultural alignment repair (extended version). In *The Semantic Web: ESWC 2014 Satellite Events - ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, pages 115–130, 2014. doi: 10.1007/978-3-319-11955-7_10.
- JÉRÔME EUZENAT. 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*, pages 185–191, 2017. doi: 10.24963/ijcai.2017/27.
- JÉRÔME EUZENAT & PAVEL SHVAIKO. *Ontology Matching, Second Edition*. Springer, 2013. ISBN 978-3-642-38720-3.
- CLAUDIA V. GOLDMAN, MARTIN ALLEN, & SHLOMO ZILBERSTEIN. Learning to communicate in a decentralized environment. *Autonomous Agents and Multi-Agent Systems*, 15(1):47–90, 2007. doi: 10.1007/s10458-006-0008-9.
- MICHAEL HUTH & MARK DERMOT RYAN. *Logic in computer science - modelling and reasoning about systems (2. ed.)*. Cambridge University Press, 2004.
- YANNIS KALFOGLOU & MARCO SCHORLEMMER. Ontology mapping: The state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.
- LOREDANA LAERA, IAN BLACOE, VALENTINA A. M. TAMMA, TERRY R. PAYNE, JÉRÔME EUZENAT, & TREVOR J. M. BENCH-CAPON. Argumentation over ontology correspondences in MAS. In *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*, page 228, 2007. doi: 10.1145/1329125.1329400.
- ELISA MARENGO, MATTEO BALDONI, CRISTINA BAROGLIO, AMIT K. CHOPRA, VIVIANA PATTI, & MUNINDAR P. SINGH. Commitments with regulations: reasoning about safety and control in REGULA. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3*, pages 467–474, 2011.

- STEFANO MARIANI & ANDREA OMICINI. Coordinated activities and change: An event-driven architecture for situated MAS. *Engineering Applications of Artificial Intelligence*, 41:298–309, 2015.
- MARCO MONTALI. *Specification and Verification of Declarative Open Interaction Models - A Logic-Based Approach*, volume 56 of *Lecture Notes in Business Information Processing*. Springer, 2010. ISBN 978-3-642-14537-7. doi: 10.1007/978-3-642-14538-4.
- ANDREA OMICINI, ALESSANDRO RICCI, & MIRKO VIROLI. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17:432–456, 2008.
- JEFFREY PENNINGTON, RICHARD SOCHER, & CHRISTOPHER D. MANNING. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.
- MAJA PESIC & WIL M. P. VAN DER AALST. A declarative approach for flexible business processes management. In *Business Process Management Workshops, BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006, Proceedings*, pages 169–180, 2006. doi: 10.1007/11837862_18.
- MAJA PESIC, HELEN SCHONENBERG, & WIL M. P. VAN DER AALST. DECLARE: full support for loosely-structured processes. In *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 15-19 October 2007, Annapolis, Maryland, USA*, pages 287–300, 2007. doi: 10.1109/EDOC.2007.14.
- DUC Q. PHAM & JAMES HARLAND. Temporal linear logic as a basis for flexible agent interactions. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007). May 14–18, 2007. Honolulu, Hawai'i, 2007*.
- GABRIELLE SANTOS, VALENTINA A. M. TAMMA, TERRY R. PAYNE, & FLORIANA GRASSO. A dialogue protocol to support meaning negotiation.: (extended abstract). In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 1367–1368, 2016.
- SANDIP SEN & STÉPHANE AIRIAU. Emergence of norms through social learning. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 1507–1512, 2007.
- NUNO SILVA, PAULO MAIO, & JOÃO ROCHA. An approach to ontology mapping negotiation. In *Integrating Ontologies '05, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Banff, Canada, October 2, 2005*, 2005.
- MUNINDAR P. SINGH. A social semantics for agent communication languages. In *Issues in Agent Communication*, pages 31–45, 2000. doi: 10.1007/10722777_3.

- JEFFREY MARK SISKIND. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61(1-2):39–91, oct 1996. doi: 10.1016/s0010-0277(96)00728-7.
- LUC STEELS. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):169–194, 1998. doi: 10.1023/A:1010002801935.
- WIL M. P. VAN DER AALST & MAJA PESIC. Specifying and monitoring service flows: Making web services process-aware. In *Test and Analysis of Web Services*, pages 11–55. "Springer Berlin Heidelberg", 2007.
- JURRIAAN VAN DIGGELEN, ROBBERT-JAN BEUN, FRANK DIGNUM, ROGIER M. VAN EIJK, & JOHN-JULES CH. MEYER. 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*, pages 899–906, 2006. doi: 10.1145/1160633.1160794.
- JUN WANG & LES GASSER. Mutual online ontology alignment. In *OAS'02 Ontologies in Agent Systems, Proceedings of the AAMAS 2002 Workshop*, volume 66 of *CEUR Workshop Proceedings*, 2002.
- FEI XU & JOSHUA B TENENBAUM. Word learning as bayesian inference. *Psychological Review*, 114(2):245, 2007.