Vocabulary Alignment for Agents with Flexible Protocols

Paula CHOCRON, Marco SCHORLEMMER
[a] *IIIA-CSIC, Bellaterra, Catalonia, Spain*

Abstract.
In recent work we presented an interaction-based alternative to the problem of vocabulary alignment in multi-agent systems. Previous approaches required external knowledge or a previously shared meta-language to fid a translation between the vocabularies of heterogeneous agents. Instead, we assume that agents share the knowledge of how to perform the tasks for which they need to collaborate, and we show how they can learn alignments from repeated interaction. The result is an online, on-demand alignment method, that uses only information agents need to interact. Until now, we have always required agents to share the complete procedural knowledge of the task. In this paper we present an extension that allows us to consider, in a meaningful way, differences between the agents' specifications. To this aim, we propose a new kind of protocols with constraints that have weights, to represent a punishment that is received when they are violated. We adapt the learning techniques from previous work to these new protocols, and present preliminary experimentation for two possible uses.

1. Introduction

The problem of aligning vocabularies has been identified as particularly relevant in the context of dynamic and open environments such as multi-agent systems [5]. Participants in this kind of systems have different backgrounds and may therefore not share the same vocabulary, making alignments necessary to interact meaningfully with each other. This problem has been considered in the past decades from two perspectives. Some approaches [9,4] assume the existence of external elements, such as physical objects, that all agents perceive in common, and explore how those can be used to explain the meaning of words. A second group of techniques [7,8] provide explicit ways of learning or agreeing on an alignment, such as argumentation techniques, explanations, or definitions. These techniques always require agents to share a meta-language. The question of how to communicate with heterogeneous interlocutors when neither a physical context nor a meta-language are available remains practically unexplored.

In recent work [1,2,3] we proposed a novel approach to the problem of multi-agent vocabulary heterogeneity. In our techniques, that we call *interaction-based*, the alignment is performed considering only the context given by the interactions in which agents engage. In this work we assume the interacting agents share the knowledge of how to perform a task, or, more concretely, the specification of an interaction. As an example, consider an *ordering drinks* interaction between an English speaking customer and an

Italian waiter. Since the agents share the dynamics of the conversation they will, for example, both know that the waiter will ask for the colour if wine is ordered. However, the words used by each agent can be different (*vino* and *colore* instead of *wine* and *colour*). In previous work, we developed techniques that agents can use to learn progressively, from the experience of interacting, which mappings lead to successful interactions. After several dialogues, agents learn an alignment that they can use to order and deliver drinks with that particular interlocutor.

An important restriction that we (until now) imposed on agents is that they have to share the entire structure of the interactions they perform. This means that they need to agree on exactly how each interaction should be performed, modulo an alignment. This is a reasonable decision considering that our goal was to study how agents can learn alignments from shared interaction specifications. However, it is in practice a strong restriction, and it always raises the question of what agents can learn if the protocol is not completely shared. Our answer was that if the differences are small, they will be ignored by the learning methods, which work statistically. If, instead, there are significant differences, agents have nothing to learn, since there is no alignment that is useful to perform the tasks together. However, we have never performed a systematic analysis of how these differences affect the learning process.

In this paper we present an extension to the problem we studied in previous work, that considers the case when agents do not share the complete alignment. To this aim, we present a new kind of interaction specifications, based on the LTL constraint protocols that we used in [3]. These new protocols include a weight for each interaction rule, introducing a meaningful way to include differences between specifications. We provide an adaptation of our learning techniques to these protocols, and a way of measuring the quality of an alignment. These ideas can be seen from two points of views. One is to consider agents that use unrelated protocols, and to analyse how the techniques can find the alignment that minimize the punishment. Alternatively, we can suppose agents use protocols that have a shared portion and then diverge in some local rules, and use the method to find the alignment for the common part. In section 4 we present preliminary experiments for these two applications.

2. Vocabulary Alignment in (Rigidly) Openly Specified Interactions

In this section we summarize the approach that we proposed in [3] to let agents infer alignments from the experience of interacting. We consider two agents a_1 and a_2 that have different vocabularies V_1 and V_2 respectively. Agents interact repeatedly using, for each interaction, the specification defined in a protocol. In this work we used open specifications consisting of rules that constrain what can be said without specifying completely the flow of the interaction. We chose to use a reduced version of the *ConDec* protocols [6], which define the constraints in linear temporal logic (LTL).

To define the protocols formally, let a vocabulary V be a set of words and consider a pair of agent IDs $A = \{a_1, a_2\}$. The set of messages $M_{V,A}$ is the set of all tuples $\langle a_i, v \rangle$ where $v \in V$ and $i \in \{1,2\}$. An interaction protocol over V and A is a tuple $\langle M_{V,A}, C \rangle$, where C is a set of constraints on how messages can be uttered. Constraints are LTL sentences renamed conveniently, that are shown in Table 2[1]. In this table, $n \in \mathbb{N}$, $a, b \in M_{V,A}$, and $\Diamond, \Box, \circ$ are the LTL operators that mean *eventually*, *globally* and *next* respectively. These constraints are generally divided into two classes. Existential constraints (*absence*) predicate over how many times a message can be uttered, while relational constraints describe binary relations between two utterances. In a protocol $\langle M_{V,A}, C\rangle$, the constraints C needs to be a subset of the possible constraints over $M_{V,A}$.

Table 1. LTL definitions of constraints

Constraint	LTL meaning
$absence(a)$	$\neg\Diamond a$
$!correlation(a,b)$	$\Diamond a \rightarrow \neg\Diamond b$
$!response(a,b)$	$\Box(a \rightarrow \neg\Diamond b)$
$!before(a,b)$	$\Box(\Diamond b \rightarrow \neg a)$
$premise(a,b)$	$\Box(\circ b \rightarrow a)$
$!premise(a,b)$	$\Box(\circ b \rightarrow \neg a)$
$imm\_after(a,b)$	$\Box(a \rightarrow \circ b)$
$!imm\_after(a,b)$	$\Box(a \rightarrow \circ\neg b)$

During an interaction, a_1 and a_2 have protocols $\mathfrak{P}_1 = \langle M_{A,V_1}, C_1\rangle$ and $\mathfrak{P}_2 = \langle M_{A,V_2}, C_2\rangle$ respectively, and each agents send messages to its interlocutor respecting the constraints in its protocol. As an example, consider again the ordering drinks scenario where the waiter w speaks Italian and the customer c speaks English. In this situation, agents could be using the following the following two protocols $\mathfrak{P}_w$ and $\mathfrak{P}_c$, where w is the waiter and c is the customer.

$$\begin{gathered}\mathfrak{P}_w = \{premise(\langle w, da\ bere\rangle, \langle c, birra\rangle),\\ premise(\langle w, da\ bere\rangle, \langle c, vino\rangle),\\ !response(\langle c, birra\rangle, \langle w, colore\rangle),\\ imm\_after(\langle c, vino\rangle, \langle w, colore\rangle)\}\end{gathered}$$

$$\begin{gathered}\mathfrak{P}_c = \{premise(\langle w, to\ drink\rangle, \langle c, beer\rangle),\\ premise(\langle w, to\ drink\rangle, \langle c, wine\rangle),\\ !response(\langle c, beer\rangle, \langle w, color\rangle),\\ imm\_after(\langle c, wine\rangle, \langle w, color\rangle)\}\end{gathered}$$

A sequence of messages is called an *interaction*. We say an interaction is *possible* for a protocol if it does not violate any of its constraints, or, more formally, if it is an LTL-model of each of its constraints.

As we mentioned, until now we required agents to share the complete structure of the protocols they use for an interaction. To formalize this restriction, we defined the notion of *compatibility* between protocols. Intuitively, two protocols are compatible if the interactions that are possible for each protocol are equivalent under an alignment. Given two vocabularies V_1 and V_2, an alignment from V_2 to V_1 them is a partial function

[1] The number of constraints in the original paper was longer, but we only include here the ones that we actually used in the learning method (the non-monotonic ones).

$\alpha : V_2 \to V_1$. An interaction in vocabulary V_2 can be translated to an interaction in V_1 using α if all its messages are in the domain of the alignment, by translating each word.

Definition 1 *Consider two interaction protocols* $\mathfrak{P}_1 = \langle M_{A,V_1}, C_1\rangle$ *and* $\mathfrak{P}_2 = \langle M_{A,V_2}, C_2\rangle$, *and let* $Int(\mathfrak{P}_i)$ *be the set of possible interactions in* V_i. $\mathfrak{P}_1$ *and* $\mathfrak{P}_2$ *are* compatible *if there exists a partial function* $\alpha : V_2 \to V_1$ *such that*

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

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

Since agents send messages composed of their own words, when they receive a foreign message they need to interpret it in their local vocabulary to continue the interaction. The objective of the learning techniques we propose in [3] is to infer the alignment α under which the protocols are compatible, something that would allow agents to interpret correctly the messages they receive.

The approach we present for learning alignments from interactions is simple. From now on we will use the point of view of a_1, but everything is analogous for its interlocutor. The agent maintains a confidence distribution $\omega : V_2' \times V_1 \to [0,1]$ that assigns a value to each mapping between a known foreign word ($V_2' \subseteq V_2$) and a word in their vocabulary. These values are updated according to what agents observe in interactions. We present here only the simplest one of the methods we proposed in [3]. Briefly, when an agent receives a word, it punishes all interpretations that are not possible because they violate some constraint. Suppose a_1 receives v_2 after observing interaction i, and let r be a punishment parameter and $i.m$ the operation of appending a message m to i. The update is described in Eq. (1).

$$\omega(v_2, v_1) := \begin{cases} \omega(v_2, v_1)\cdot(1-r) & \text{if } i\,.\langle a_2, v_1\rangle \text{ is not possible for } \mathfrak{P}_1 \\ \omega(v_2, v_1) & \text{otherwise} \end{cases} \tag{1}$$

For example, if the customer receives *colore* right after saying *wine*, it infers that it cannot mean *to drink*. By interacting repeatedly with different protocols, agents gradually learn an alignment between their vocabularies.

3. Flexible Protocols

We now propose an approach that considers more carefully the question of to what extent agents can align their vocabularies when their protocols are different. To this aim, we introduce *flexible protocols*, in which each constraint has a weight that represents a punishment received when that constraint is violated. This punishment can be interpreted, for example, as a way of expressing preferences (heavier constraints are those that agents prefer not to violate), or degrees of confidence on a constraint, when there is uncertainty about the interaction context.

Definition 2 *A flexible protocol $\mathfrak{P}^f$ over a vocabulary V and a set of agents A is a tuple $\langle M_{V,A}, C^f \rangle$. The set C^f is composed of pairs $\langle c, \rho \rangle$, where c is one of the constraints in Table 2 over $M_{V,A}$, and $\rho \in [0,1]$.*

As an example, consider again the ordering drinks scenario. Assume they have the same constraints as before with high weight, but now the waiter also believes that the customer should not order two different alcoholic beverages in one interaction. This constraint, however, is less strict than the others, since the waiter is willing to accept that behaviour some times. The protocols would look as follows.

$$\begin{gathered}
\mathfrak{P}_w = \{\langle \mathit{premise}(\langle w, \mathit{da\ bere} \rangle, \langle c, \mathit{birra} \rangle), 1 \rangle, \\
\langle \mathit{premise}(\langle w, \mathit{da\ bere} \rangle, \langle c, \mathit{vino} \rangle), 1 \rangle, \\
\langle !\mathit{response}(\langle c, \mathit{birra} \rangle, \langle w, \mathit{colore} \rangle), 1 \rangle, \\
\langle \mathit{imm\_after}(\langle c, \mathit{vino} \rangle, \langle w, \mathit{colore} \rangle), 1 \rangle, \\
\langle !\mathit{correlation}(\langle c, \mathit{vino} \rangle, \langle c, \mathit{birra} \rangle), 0.5 \rangle\}
\end{gathered}$$

$$\begin{gathered}
\mathfrak{P}_c = \{\langle \mathit{premise}(\langle w, \mathit{to\ drink} \rangle, \langle c, \mathit{beer} \rangle), 1 \rangle, \\
\langle \mathit{premise}(\langle w, \mathit{to\ drink} \rangle, \langle c, \mathit{wine} \rangle), 1 \rangle, \\
\langle !\mathit{response}(\langle c, \mathit{beer} \rangle, \langle w, \mathit{color} \rangle), 1 \rangle, \\
\langle \mathit{imm\_after}(\langle c, \mathit{wine} \rangle, \langle w, \mathit{color} \rangle), 1 \rangle\}
\end{gathered}$$

3.1. Updating Technique

The core contribution of our previous work consists in the learning techniques that we proposed to update the agent's mapping confidence distribution. In rigid protocols the objective of the learning is to find a correct alignment α under which the protocols were compatible. In [3] we show how, to that aim, it is enough to consider whether a word is a possible interpretation for a received message. The notion of compatibility cannot be applied to a flexible protocol, since the question of whether an interaction is possible is not meaningful anymore. When interacting with a flexible protocol, any sequence of messages can be accepted, but at some cost. This implies that there is no reference alignment α that agents can use to understand each other completely. Instead, each alignment will have an expected punishment that will make it better or worse. The goal now is to find an alignment that optimizes the punishment that is received while interacting.

To do so, agents need to take into account the weight of the rules that would be violated for each interpretation. We propose only one simple approach, that combines the punishment that would be received in a particular moment as well as the information that the agent already had.

As in the rigid cae, agents that use flexible protocols have a confidence distribution ω that assigns a value to each possible mapping. Consider again agents a_1, a_2 with flexible protocols $\mathfrak{P}^{\mathfrak{f}}_{\mathrm{j}} = \langle M_{V_i,A}, C^f_j \rangle$ for $j \in \{1,2\}$. Suppose a_1 receives a word v_2 from a_2 after interaction i. For a word $v \in V_1$, let $B(v, \mathfrak{P}^{\mathfrak{f}}_1)$ be the set of all constraints c included in a tuple in C^f_1 such that the interaction obtained by appending $\langle a_1, v \rangle$ to i violates c, but i does not. Let $\rho(\mathfrak{P}^{\mathfrak{f}}, c)$ be the weight of constraint c in $\mathfrak{P}^{\mathfrak{f}}$. Then, for each $v \in V_1$, and for each $br \in B(v, \mathfrak{P}^{\mathfrak{f}}_1)$, agent a_1 iteratively updates:

$$\omega(v_2, v) := \omega(v_2, v) * (1 - \rho(\mathfrak{P}^{\mathfrak{f}}{}_1, br))$$

After all updates are done, the values are normalized to make $\sum_{v \in V_1}(\omega(v_2, v)) = 1$.

3.2. A Distance Measure

An important question to address before adapting our techniques to the flexible case is the one of when an alignment is good for these new protocols. In the rigid case we measured the quality of an alignment by computing its distance to the reference alignment α, using the standard precision and recall measures. In flexible protocols, where the objective is to minimize the received punishment, a quality measure should reflect how much agents would pay when using them.

We propose a method that evaluates a pair of alignments for a given pair of protocols. Alignments are evaluated in pairs, and not by themselves, because the punishment that an agent receives depends on how its interlocutor interprets the words it utters. The objective is, therefore, to measure how well they would interact together if using the obtained alignments.

Our objective will be to measure the quality of the alignments that agents learn during with our techniques. Since agents have only confidence distributions, we first need to explain how to obtain an alignemnt from them. For simplicity, in this explanation we will assume that these alignments are, for agent a_1, a partial function $\alpha_1 : V_2 \rightharpoonup V_1$. This alignment maps each foreign word known by the agent with the local word for which the mapping value ω is maximal. In practice, agents frequently do not have enough information to distinguish between possible mappings, so there are many of them with maximal value. Therefore the alignment is actually a function $\alpha_1 : V_2 \rightharpoonup \mathscr{P}^{V_1}$, where $\mathscr{P}^S$ is the set of all subsets of a set S. To adapt the measure we present now to this kind of alignments it is only necessary to consider α_1 and its analogue for a_2, α_2, as a set of possible alignments, and compute the measure for each combination.

The straightforward way to compute the expected punishment for a pair of protocols and a pair of alignments consists in obtaining all interactions, filtering the ones that could actually occur in an interaction, and computing the average of the received punishment for them. Even if the interactions have bounded length, this method is computationally very expensive, since the amount of interaction grows exponentially with the length. Fortunately, for our particular set of constraints, it is enough to consider only the punishment received by interactions of length 2 to have a measure of the alignment's quality. This is because the constraints are unary or binary, so any violation that could happen would also occur in an interaction of length 2. In future work we plan to investigate in more depth the relation between the measure computed in this way and the actual expected punishment for longer interactions.

We consider only interactions that could be uttered, and discard all sequences of messages that would never occur in a real exchange. These impossible interactions are the ones in which an agent sends an utterance that causes an unnecessary punishment. The notion of *utterable* interactions is defined over *abstract* interactions. These are sequences that are actually said before any interpretation, where the messages uttered by a_1 have words in V_1 and the ones by a_2 have words in V_2.

Definition 3 *A two-message abstract interaction* $i = [\langle a_j, w\rangle, \langle a'_j, w'\rangle]$, *for* $j \in \{1,2\}$ *is* utterable *for alignments* α_1, α_2 *and protocols* $\mathfrak{P}_1$ *and* $\mathfrak{P}_2$ *if:*

- $[\langle a_j, w\rangle]$ *does not violate any constraint in* $\mathfrak{P}_j$ *and*

- $\alpha'_j(i)$ *does not violate more constraints for* $\mathfrak{P}'_j$ *than* $\alpha'_j([\langle a_j, w\rangle])$ *does.*

We will call $U_{\mathfrak{P}_1,\mathfrak{P}_2,\alpha_1,\alpha_2}$ to the set of all utterable sentences for those protocols and alignments. Next we need to define the punishment for an interaction with a given alignment and protocol.

Definition 4 *The punishment for an interaction i and a protocol* $\mathfrak{P}$ *(noted* $\rho(\mathfrak{P}, i)$*) is the added weight of all constraints that are violated by i in* $\mathfrak{P}$.

The quality of alignments α_1, α_2 for $\mathfrak{P}_1, \mathfrak{P}_2$ is:

$$\frac{\sum_{i \in U_{\mathfrak{P}_1,\mathfrak{P}_2,\alpha_1,\alpha_2}} \rho(\mathfrak{P}_1, \alpha_1(i)) + \rho(\mathfrak{P}_2, \alpha_2(i))}{|U_{\mathfrak{P}_1,\mathfrak{P}_2,\alpha_1,\alpha_2}|}$$

This quality measure is bilateral and measures the amount paid by two agents. It can also be made unilateral by only considering the punishment for one protocol, but it will still depend on both alignments. Note also that we defined the measure for a case in which all words have the same likelihood of being uttered, but it can be easily adapted for other cases. If we know the probability of uttering a given word, to take it into account in the measure it is enough to multiply the punishment for an interaction $i \in U_{\mathfrak{P}_1,\mathfrak{P}_2,\alpha_1,\alpha_2}$ by the likelihood of i. The extension is not so simple when agents have more complicated distributions to choose an utterance, such as one in which the likelihood of sending a message depends on the previous interaction.

4. Experimental Evaluation

Flexible protocols, along with the updating technique that we proposed, can be analysed from two points of view. First, we can consider agents that have structurally different protocols for the same task, without requiring any a priori relation between them. In this case, the technique can be used as a tool to find an alignment that minimizes the punishment received by agents for that specific protocol. This can also be used for small sets of protocols.

A second point of view considers flexible protocols as local variations of a common compatible part. In this case we study how the technique lets agents find the correct alignment (the one that is useful for the common par) in spite of the differences. In the following sections we explain the experiments that we performed to study these two cases.

4.1. Learning an Alignment that Optimizes Punishment

In this set of experiments we analysed how our method can be used by agents that have different protocols, without any similarity assumption. The objective of the interacting agents is to find the alignment that would make them pay as little as possible. For each experiment we created a pair of flexible protocols, gave one to each agent, and let them interact for some time, observing the alignment that they obtained in the end. To decide the quality of the alignments, we used two different measures:

1. The real punishment received by agents when they used the alignment

2. The quality of the alignments with respect to the protocols measured as explained in Section 3.2

Since we proposed the alignment quality measure as an expected value of the punishment received by agents, our hypothesis is that the curves obtained with it are similar to the ones that show the average received punishment. However, the distance measure only considers the expected punishment for interactions of length 2. Since agents use longer interactions, the value of the real punishment will be larger.

An experiment consists of two agents, each of them with a vocabulary and a protocol over it. In this preliminary experimentation the size of the vocabularies is of 4 words. We performed experiments with different protocol sizes and total weight (obtained by adding up the weight for each constraint). Concretely, we used protocols with 6 and 8 rules, and total weight 3.0 and 5.0. In each experiment, both protocols have the same number of constraints and the same added weight. In an experiment, we let agents go through a learning phase in which they interacted n times, performing the experiment for $n = [0, 2, 5, 10, 15, 20, 30]$. During these alignments they updated their confidence distributions on mappings. All interactions were of length 6. After this training phase, we ran a test phase of 20 interactions, during which agents did not learn new information. Instead, they interpreted foreign words using the alignments obtained in the previous phase. For the second measure, we added the total punishment that agents received during the test phase. We repeated the experimentation 10 times, for 10 different protocols.

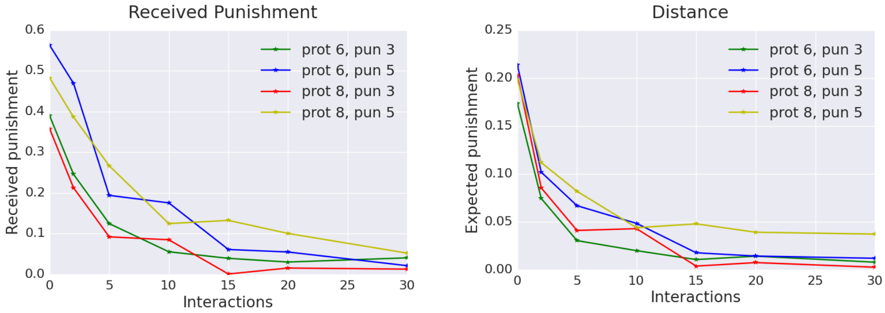

Figure 1. Results for a vocabulary of 4 words

Figure 1 shows the results obtained for each experiment. We can see that with more training, agents find alignments that are better according to the measure, and that let them interact receiving lower punishment. The distance measure seems to change in the same way as the actual received punishment, which indicates it is a good estimate. We would need a larger set of experiments to answer some interesting questions that are not reflected in this data, such as the relative importance of the amount of constraints and the total weight.

4.2. Learning a Correct Alignment in Protocols with Variations

The objective of the second type of experiments is to analuze the learning techniques when applied by agents with protocols that diverge in some degree from a shared compatible part. To build these kind of protocols, we used four parameters:

- r_s : the number of shared constraints
- r_d : the number of different constraints
- w_s : the total weight for shared constraints
- w_d : the total weight for different constraints

We started from two vocabularies V_1 and V_2 and an alignment α between them, and created two protocols with r_s constraints that were compatible under α. This was the shared part. We distributed the weight w_s over those constraints (to simplify the experimentation, we did this uniformly). Then, we added r_d different constraints to each protocol, and distributed the weight w_d between them (not necessarily uniformly).

In each experiment, we let agents interact 100 times with different pairs of protocols, and measured when (and if) they converged to α. It is important to make one comment here. In our previous work, it was enough to consider that agents had found α when their learned alignments were equivalent to it. This is because in rigid protocols, using α would always lead to correct interactions. This is not the case any more, since agents could receive punishments even using α, that could made them change the alignment. For this reason, we considered an agent had converged when after they had spent 15 interactions without changing the alignment (but we recorded the number where they first got to α, before those 15 interactions).

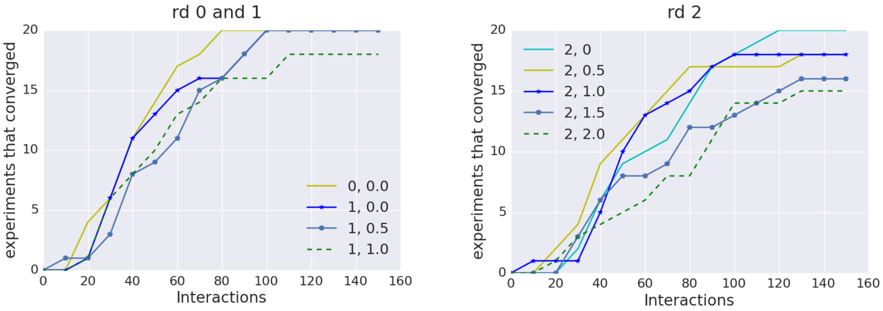

Figure 2. Results for a vocabulary of 4 words and protocols with 8 common constraints

In our experiments, we considered a large set of shared constraints (8 constraints, with weight 0.8 each), and studied the convergence to α for different values of r_d and w_d The results can be seen in Figure 2. We performed the experiment 10 times, with different protocols. The figure shows the number of experiments after each amount of interactions. We used $r_d = \{0,1,2\}$ and $w_d = \{0.0,0.5,1.0,1.5,2.0\}$. Some combinations were not performed, because the weight was impossible to distribute in the amount of rules. We see how more and more heavy different constraints imply more difficulty to find α. We also observe an unexpected effect. Even with the same weight, more constraints slow down the convergence. We think this is because of how agents choose which word to utter. Now, they only utter words that do not violate any rule, so they are more restricted, which could make interactions finish in failure earlier, or could make all interactions very similar between them. Another thing to comment is that the difference between small weights cannot be seen correctly in these experiments (for example, see the case od $rd = 2, wd = 0.0$ vs $rd = 2, wd = 0.5$). We plan to study this effect more exhaustively in the future, with a larger set of experiments.

To have a deeper comprehension of how these techniques work, they should be compared to the performance of the original learning methods for rigid protocols, when used for protocols that have some diverging constraints. In this way, we would be able to detect if taking into account explicitly the weight of different constraints is better than considering them as noise.

5. Conclusions

We introduced a new kind of protocols that allow us to investigate the problem of vocabulary alignment for agents that do not share exactly the same procedural knowledge of the tasks they perform together. This can mean either that they have unrelated specifications, or that they share a common part that has a significant number of constraints and another portion differs. The technique that we propose to learn alignments in this case is a simple adaptation of the one we had introduced in previous work, and is useful for both situations.

This work is still in a preliminary state, but we see it as a basis to many extensions. First of all, the experimental part needs to be improved, considering larger protocols and making a more accurate analysis. The current experiments should only be seen as preliminary tests that give an idea of how the method works and provide input for future experimentation. For example, in the first experiment it is possible to consider the optimality of the solution the agents find, something that would be useful to understand how the technique works. In addition to this, there are different questions to consider regarding the technique. We think it would be particularly interesting to consider interactions between agents with different kinds of preferences. For example, what kind of alignment would a pair of agents reach if one of them has very high preference values and the other one very low?

References

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

[2] Paula Chocron and 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.

[3] Paula Chocron and Marco Schorlemmer. Vocabulary alignment in openly specified interactions. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017) (To appear)*, 2017.

[4] Jurriaan Van Diggelen, RobbertJan Beun, Frank Dignum, Rogier M. Van Eijk, and John Jules Meyer. Ontology negotiation: Goals, requirements and implementation. *Int. J. Agent-Oriented Softw. Eng.*, 1(1):63–90, April 2007.

[5] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[6] M. Pesic and W. M. P. van der Aalst. A declarative approach for flexible business processes management. In *Proceedings of the 2006 International Conference on Business Process Management Workshops*, BPM'06, pages 169–180, Berlin, Heidelberg, 2006. Springer-Verlag.

[7] Gabrielle Santos, Valentina Tamma, Terry R. Payne, and Floriana Grasso. A dialogue protocol to support meaning negotiation.: (extended abstract). In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '16, pages 1367–1368, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

[8] Nuno Silva, Gecad Isep Ipp, and Paulo Maio. An approach to ontology mapping negotiation. In *Proceedings of the Workshop on Integrating Ontologies*, pages 54–60, 2005.

[9] Luc Steels. The Origins of Ontologies and Communication Conventions in Multi-Agent Systems. *Autonomous Agents and MultiAgent Systems*, 1:169–194, 1998.