A Computational Model for Mutual Intelligibility in Argumentation-Based Multiagent Systems

by

KEMO ADRIAN

A dissertation presented in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy in Computer Science

Advisor:
Prof. Enric Plaza Cervera

Tutor:
Prof. Enric Plaza Cervera

November 28, 2019

Universitat Autònoma de Barcelona
Departament de Ciències de la Computació

Abstract

In supervised symbolic learning, examples are related to signs through strict associations. These associations, given by a third party, are considered as fixed by intelligent systems that received them. Agent systems with learning capabilities inherit this strict assumption on their example-sign associations. This thesis presents a different approach to example-sign associations for multiagent learning systems, where the example-sign associations are instead fluid and adaptive, being able to evolve during communication between two agents. We believe, in fact, that having strong assumptions about the signs associated with examples prevents agents to effectively communicate in situations of semantic heterogeneity. Our approach models elements from the fields of semiotics and anthropology in order to allow the agents of a multiagent system to dynamically change their example-sign associations, and therefore, when they observe disagreements in situations of semantic heterogeneity be able to resolve them and reach mutual intelligibility.

This research work is presented into five stages. First, we introduce the problem of reaching mutual intelligibility in scenarios where disagreements are observed. The second stage is introducing a semiotic viewpoint that characterizes our approach and allows agents to communicate on their example-sign associations. The third stage is the presentation of an argumentation model that assumes error-free concept learning. The fourth stage is extending this model to an error-tolerant argumentation model, which can reach mutual intelligibility while assuming a certain degree of error in concept learning. The fifth stage is the presentation of two strategies adopting our approach and our argumentation model: the systematic and the lazy strategy. The systematic strategy is one where agents, upon meeting, start arguing about their concepts, in order to resolve their disagreements all at once. The lazy strategy considers two agents resolving disagreements one by one, as they arise in their interaction. We experimentally evaluate the performances of our error-tolerant argumentation model, using both argumentation strategies, and show that agents using our approach can resolve any disagreement, or combinations of them, while increasing their mutual intelligibility. Moreover, we show that the agents are able to resolve their disagreements and improve their mutual intelligibility in several application domains. Finally, we show that our argumentation model does not require extensive amount of information exchange between agents to attain the state of mutual intelligibility.

Acknowledgements

After five years, this journey that is a thesis comes to an end. It has been, by many aspects, both a marathon and a sprint. This time taught me so much, both on an academic and on a personal level, and it leaves me with so many acknowledgments to give. Looking back to all the work done, I cannot help but think that this achievement belongs to all the people that have helped it come into being.

First and most of all, I would like to thank Professor Enric Plaza for his supervision. He has been my tutor during these past five years and has put the most efforts in the writing and editing process. Additionally to the invaluable knowledge that I acquired from him, he taught me patience and persevering, dedication to work, and that all hard things can come to a good end. I would like to thank Professor Fiona McNeill who welcomed me to the University of Edinburgh and found time to help me through a Scottish winter full of ontologies.

My very special gratitude goes out to Professor Michael Rovatsos, without whom the ESSENCE Network would not have been what it was, and neither would I. All of the people in this network have been there for me, teaching me and supporting me. Some special mentions go to Dagmar and Paula, who have done so much to see this thesis succeed, and have been an example to me. All members of the network have been more than colleagues: thank you Thomas, Paolo, Julian, Michael, Paul, Mercedes, Khuyagbaatar, Aimilios, Francisco, Aysenur and Nicola.

If Edinburgh and the ESSENCE Network have been amazing places to grow and meet new people, my main home during this time has been the Instituto de Investigación en Inteligencia Artificial. I always found people to help me on all matters that were on my mind. In this regard, I would like to thank Professor Carles Sierra for his talks on Semiotics —and his marvelous costelladas. Of course, I would like to thank the amazing colleagues and friends that went through the ordeal of a Ph.D by my side: Ewa, Xavi, Jesus, Oguz, Joan Ramon, Sergi, Broja, Toni, Albert, Andrew, Tito and Roberto. I could not forget to mention the great administrative team of the institute, in particular Ana Beltran, Montse Sanmartí and Montse Calderon, that saved me many times from the attacks of Spanish administration. I extend this acknowledgement to all the people that are or have been my colleagues, as they created an atmosphere that made me feel at home.

This acknowledgement could not be complete without giving the warmest thanks to my parents, Sylvie and Philippe, that have always been on my side, years before I started even thinking of doing a Ph.D. Thank you for all the answers to my stupid questions asked. As you can see, I did not stop asking them. Thank you Quentin, my brother, for thinking of me as an example. You made me be the man that you have always seen in me. And finally, I would like to thank my Fiancé. She has been the emotional cornerstone of this thesis. Thank you, Irina. For your patience, your kindness and your everyday support. Thank you so much.

This work has been partially supported by projects ESSENCE: Evolution of Shared Semantics in Computational Environments (ITN 607062) and DIVERSIS (CSIC Intramural 201750E064).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis presents an argumentation model of argumentation that allows two agents to to reach mutual intelligibility in situations of semantic heterogeneity. We have based our model on a semiotic approach to meaning, where the agents can question the relation of particular symbols with both their intensional definitions and their extensional definitions, in order infer relations between their concepts which help them to create a new vocabulary that improves mutual intelligibility. This approach differs in several interesting ways from the traditional approach of symbolic machine learning, in which examples from classes and generalizations learned open these classes are given a particular sign as a label that cannot be modified. Our argumentation model aims to take the possibility of loose example-sign relations that can be changed in order to understand certain types of disagreements during communications in multiagent systems. Moreover, our argumentation model aims at creating a *contextual* meaning of concepts between two agents of a system, that is able to reach the mutual intelligibility between both agents within the context of their current domain of interaction. Indeed, we believe that in a large number of situation, requiring agents to have a predefined consensual meaning on the whole system is too strong and sometimes not achievable, while we claim that addressing contextual meaning, we can reach consensual or mutually agreed meaning over the specific interactions that they have with a restricted set of interlocutors. Of the general model drawn from this approach, we propose two different strategies of argumentation to reach mutual intelligibility. The first strategy is a systematic strategy, in which the agents perform an argumentation process over the meaning of all their contextually relevant concepts as soon as they meet, in order to guarantee mutual intelligibility in future communications. The lazy strategy is a second strategy that works on demand and is problem-centered: when a disagreement arises between two agents the engage in an argumentation process until that disagreement is resolved and (partial) mutual intelligibility is reached.

1.1 Motivation

In supervised symbolic learning, examples are tied to signs through a strict association that is considered as given by an oracle and cannot be revised. In approaches combining multiagent systems where agents have symbolic learning capabilities, the agents inherit the strict constraint in the assumptions taken about the example-sign relation from the symbolic learning paradigm. This thesis present a different approach to example-sign relations in learning multiagent systems, where these relations are instead fluid, adaptive and evolving during the communication process of two agents. We believe that having strong assumptions on the sign that should be given to an example can, in certain scenario, prevent the agents to effectively communicate. Our approach provides a model that allows the agents to modify their example-sign relations in order to reach a state of disagreement-free communication, called *mutual intelligibility*, in scenarios where, with strict example sign relations, these problems could not solved by the agents themselves. Mutual intelligibility can be regarded as agreement over the meaning of concepts. The definition of what is "meaning" has been and continues to be debated in many fields, from philosophy of language, to linguistics and natural language processing. We do not intend to give a resolution to these debates, instead we will take a specific viewpoint on the notion of meaning and we will show how it can be useful to the main goal of our work. In order to stay consistent with our scope to reach mutual intelligibility through the modification of example-sign relations, we present the meaning of a concept as the different solutions available to an agent in order to associate a particular example with a particular sign in a manner that can be mutually consensual with another agent —within the specific context that they share.

This approach is consistent with the work done in linguistics and language philosophy to understand how disagreeing speakers of a same natural language often do not disagree on the content (examples or generalizations of these examples), but on the words (signs) used to express that content. In computer science, this phenomenon is often referred as semantic heterogeneity when two different parties create data-sets for the same domain that have differences in meaning and interpretation of data content. The privileged domain to study the relation between an sign and its meaning is the field of semiotics. In semiotics, the relation between symbols and the objects they represent are figured as a triangle, also known as the semiotic triangle, first proposed by by Odgen and Richard in 1923 (Ogden and Richards, 1923). This triangle is represented in Figure 1.1 (left). The semiotic triangle figures three elements; as we mentioned, the first two are a symbol and the object that this symbol is referring to (called referent). The third element is the reference, which is the meaning that a speaker associates a given object to a given symbol. These three elements are represented as the three vertexes of the semiotic triangle.

The first step taken by our computational approach is to associate each of the three elements from the semiotic triangle to a corresponding element of machine learning, in order to create a computational semiotic model also represented as a triangle. First, our approach uses the term *sign* instead of *symbol*, sign being a more generic term also used in semiotics. Then, our approach does not have objects but examples. These examples are regrouped in *extensional definitions*, that are the elements corresponding to the referent in our model. The extensional definition of a term is commonly understood as a listing of every object represented by that term. In our approach, the extensional definition of a triangle is the set of examples that are currently associated with the sign of that triangle. Finally, our approach considers the reference as an *intensional* definition. An intensional definition is a set of generalizations that can, with the use of a relation of subsumption, determine whether or not an example should be part of a specific triangle. Together, the sign, extensional and intensional definition form a computational model of semiotic triangle similar to the model introduced by Manzano et al. (Manzano et al., 2012). An intuitive way to present the sign, intensional and extensional definitions of a concept is to think about even numbers. The whole concept of even numbers can be seen as the relation between its name (sign) *even*, its intensional definition that could be formulated as: *for all natural x in $\mathbb{Z}$, x is even if x is a multiple of two*, and its extensional definition: the set of naturals $\ldots, -2, 0, 2, 4, \ldots$ that verify the intensional definition.

An explanation to why speakers use different meanings has been given in anthropology by Frake (Frake, 1962), that proposes the notion of *contrast set* to designate a set of concepts used by a speaker in a particular context or task. Frake introduces the notion of contrast set with the idea that meaning is deeply contextual. This echos the late position of Wittgenstein on meaning (Wittgenstein, 2009), where meaning can only be understood in the context of a specific interaction. Our model starts from this explanation to give concepts a meaning that does not only depend on the elements of their associated triangles, but also from the relation that their semiotic triangle has with the semiotic triangles —and therefore the meanings— of neighbouring concepts.

We see every day real word instances explaining the usefulness of contrast sets. For instance, a buyer can enter an eatery and ask "What kind of sandwiches ya got besides hamburgers and hot dogs?", to which the seller responds "How about a ham 'n cheese sandwich?". Here the collection

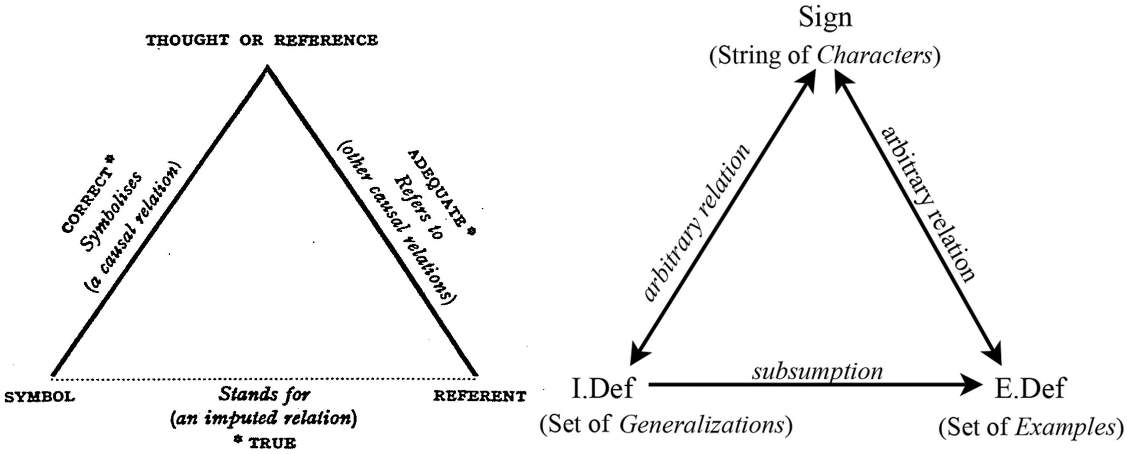


Figure 1.1: Original semiotic triangle from the work of Odgen and Richards (Ogden and Richards, 1923) (left) compared to our model of concepts represented as a triangle (right).

of words describing the different kinds of products one can eat are the contrasts set: hamburgers, hot dog, ham 'n cheese sandwich, etc. However, the way one person segregates and the word or sign used to reference them is contextual, which can lead to misunderstandings that will require, to be resolved, some adaptation of the intended meaning. An example of misunderstanding from (Frake, 1962) is the client complaining with a sentence like this: "Hey, that's no hamburger; that's a cheeseburger!". The origin of the misunderstanding is that the client, with his own customer experience in other eateries, is considering *hamburger* and *cheeseburger* as two different segments in the contrast set he is using to conceptualize the eating options, while the seller uses common culinary meanings where the extensional definition of "cheeseburger" is a subset the extensional definition of "hamburger".

As a running example of context-dependent meaning we will use the common sense domain of *Furniture Shopping*. Let's assume they have some default meaning of some concepts (often called Ontologies in Artificial Intelligence), for instance about furniture. If we ask the agents before they interact if an armchair is a chair they would probably answer affirmatively. For our purposes, we can set that armchair is a sub-concept of the chair concept. Now, imagine the buyer enters the shop and tells the seller this: "I wan to buy one armchair and four chairs". If the seller understands the meaning intended by the buyer no misunderstanding arises, and the will keep talking about "chairs" and referring to particular objects in the shop that are "chairs" without any disagreement on any specific object. And, nevertheless, they are not using "chair" as the same concept as before: now the concept chair in fact means "chairs without arms". This so because the buyer has created the contrast set {*armchair*, *chair*}, and by doing so he has (implicitly) decided to use the word "chair" with a new intended meaning. If the two agents consistently use the term "chair" to refer only to objects in the shop that are chairs and are not armchairs, we say they have achieved an agreement on meaning. This "shift" in the meaning of a term or word is so pervasive that we humans are hardly aware of it, but we would consider very wrong if the seller tried to sell three armchairs and two chairs without arms (which is consistent with the default meaning of chair and armchair). Our goal is to develop an argumentation model that allows this sort of fluid and evolving naming of concepts and objects (or situations) in agent-based systems, and that the agents by themselves would be capable of recognizing and resolving the disagreements on meaning that may arise.

Now, the issue we need to address is how to represent concept meaning in a way that allows us to have an argumentation model in which this "shift" in the meaning by creating contrast sets. The approach is semiotic, in which a *concept* is represented by a semiotic triangle $\langle S, I, E \rangle$ with three components: a sign S, a meaning (or *intensional definition*), and an object or reference

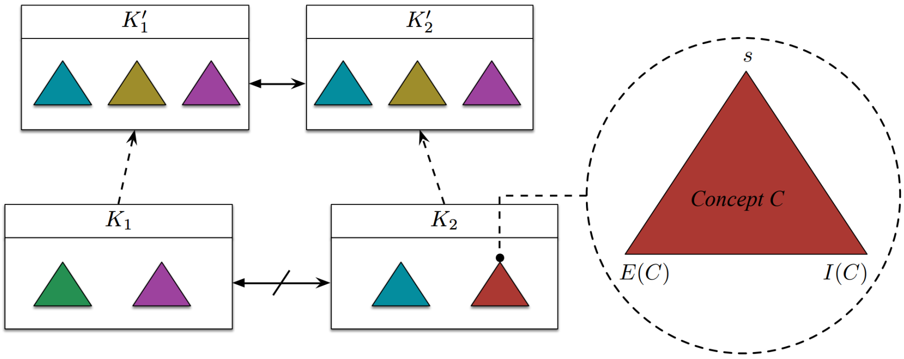


Figure 1.2: Creation of a new contrast set from two contrast sets causing disagreements.

(or *extensional definition*). In this view, a sign like "chair" can have two different meanings in the *Furniture shopping* scenario by being in two different semantic triangles. What we called the default meaning is that often found in dictionaries and ontologies, that specifies the typical or more frequent sense of a sign like "chair", and could be expressed in a semiotic triangle $\langle \textit{``chair''}, I, E\rangle$ where I is the default meaning of chairs (including armchairs and other sorts of chairs), and E is the objects that can be referred to by that sign. However, after the buyer introduces the new contrast set {armchair,chair}, the meaning of the sign "chair" needs to change. In the *Furniture Shopping* scenario, the agreed meaning of that sign can be expressed in a new semiotic triangle $\langle \textit{``chair''}, I', E'\rangle$, where now the agreed meaning I' is that of chairs without arms (because when referring to those the agents would use the "armchair" sign); moreover, the set of objects that are reference of the sign is also changed, since E' is about objects that are chairs but not armchairs.

Our approach generalizes these real word instances of interactions as naming games. Naming games are conceptual tools of language philosophy, introduced by Wittgenstein (Wittgenstein, 2009), to illustrate how meanings can emerge from the interactions of speakers. More specifically, we will present an model of two agents playing a naming game and resolving any disagreements that would occur during that game by arguing with each other and changing their example-sign relations. Specifically, our argumentation model will assume two agents with possibly different contrast sets, and we assume that each segment in a contrast set corresponds to a concept, with a semiotic triangle incorporating a sign S_1, and with the objects in a segregate corresponding to the extensional definition E_1 of that concept. Moreover, disagreements and negotiation of an agreement over meaning will be performed by an argumentation-based communication between two agents, explained in Chapter 4.

This model allows two agents in a situation of semantic heterogeneity to reach a contextual mutual intelligibility by exchanging information about their concepts in their contrast sets (and therefore their example-sign associations) in an iterative fashion that gradually increases the mutual agreement on meanings of the agents, until another contrast set that satisfies both agents is found. This process is illustrated in Figure 1.2, where two agents with concepts represented as colored triangles have initially their contrast sets K_1 and K_2 segregating different colors. The agents reach mutual intelligibility by creating a new concept in their contrast sets; adding this new concept their contrast sets become K_1' and K_2', that now segregates the objects according to their colors in a mutually agreed way. The agents, by creating a new concept and therefore changing example-sign associations, have reached mutual intelligibility through the reorganization of these associations.

1.2 Argumentation Model

This section presents the main aspects of the argumentation model that we developed in our thesis, and the approach to mutual intelligibility that this argumentation model makes of semiotics in the context of semantic heterogeneity.

Multiagent System First of all, our model is a model for multiagent systems with two agents. Agents are working by pair in order to reach a mutual intelligibility. The agents can interact with the part of their overall context that they have access to and exchange different types of messages in order to reach mutual intelligibility. Moreover, the agents can identify situations where mutual intelligibility seems to be compromised and share the information with each other. The multiagent system communication protocol is turn based, with every action of an agent being completed during a turn, and the other agent having the next turn. Only one agent can act at a given time. The actions taken by an agent during its turn depend on the messages that this agent has received, and its internal state.

Contextual Meaning In our model, the meaning is contextual. This means that our agents do not try to create concepts with a general meaning that can apply to every situation or to the classification of every possible example. Instead, the agents use their meaning to classify the examples from a particular part of their overall context that they have access to, called their context. When the agents start interacting, they are interested in extending the set of examples that they can satisfyingly classify to the context of the other agents, in order to reach mutual intelligibility. This means that rather than having an objective notion of truthfulness in their evaluation of concepts, the agents have an agreement based approach. A given concept C is satisfactory if both agents agree that C gives an individual accurate classification.

Argumentation Model We just stated that a concept is satisfactory to an agent according to individual consideration, and that the agents needed to have a mutual agreement that a concept is satisfactory. Since this satisfaction is subjective, and since an agent does not have access to the information that can allow it to represent the subjectivity of the other agent, the agents will have an argumentation in order to decide which part of a concept can be agreed on by both agents and which part needs modifications in order to reach mutual intelligibility. This expression of mutual intelligibility in terms of agreement allows to express the absence of mutual intelligibility in terms of disagreement. The disagreements are the elements that prevent at least one agent to give its agreement on a concept. The interesting thing about disagreements is that they can be listed and individually addressed to be resolved. Once all the disagreements of the two agents have been resolved, the agents will have reached mutual intelligibility.

Contrast Sets Mapping Disagreements are expressed as the result of a particular relation between two concepts. In order to characterize these disagreements, we introduce a typology of relations between pairs of concepts, called *pairing* relations (e.g. two concepts can be overlapping, or one can be a sub-concept of another). In order to identify the disagreements, we will use the pairing relations between the concepts of both contrast sets. A set of these pairing relations constitutes a mapping between the two contrast sets. Since the disagreements depend on the subjective knowledge of an agent, the pairing relations also depends on it. Therefore, each agent has an individual contrast set mapping. Since we have different pairing relations between the agents, their disagreements are characterized individually. We will present a protocol to infer overall pairing relations from the individual pairing relations.

1.3 Goals and Contributions

1.3.1 Goals of our Approach

Our aim is to provide an approach that can effectively represent new strategies for learning agents to associate their examples with their signs, allowing these agents to dynamically change their meanings when they identify disagreements in their communication. Moreover, our approach aims to provide strategies for the agents to understand which changes should be made *locally*, with minimal knowledge over each other's associations. Our approach is explained in several chapters, from Chapter 3 to Chapter 6.

1.3.2 Goals of our Model

The goals of our argumentation model are centered around reaching mutual intelligibility within between two agents. The goals can be listed in six main propositions. The experimental evaluation of each proposition phrased as a hypothesis is presented in Chapter 10, while an exemplification of our model (that illustrates in detail how our model manages to satisfy these hypotheses) is presented in Chapter 9.

Generality There can be an huge number of scenarios in which two agents face semantic heterogeneity and do not have mutual intelligibility. Our first goal is to develop a notation that allows to build a taxonomy over these scenarios, identifying which kinds of disagreements between the agents may prevent mutual intelligibility. Then, we aim at developing an argumentation model that can correctly address any arrangement or combination of disagreement and reach mutual intelligibility. Moreover, we aim at developing an argumentation model that *refines* concepts, that is to say: for every pair of examples that were differentiated in the initial contrast sets (i.e. they were in different concepts), at the end this pair of examples will not be conflated into one concept, but will still be differentiated into different concepts.

Domain Independence The agents can interact within different domains. A domain represents a set of examples that the agents can classify, and from which the agents expect a similar classification from other agents. Domains vary ontologically, and examples from different domains can have very different properties. The domain affects the ease with which the agents can classify it, as is well known from Machine Learning. In some domains, generalizations can be learned over samples that correctly classify the entire data. In some other domains, generalizations learned over some examples have difficulty in accurately classifying correctly some new unseen examples, and some degree of error is unavoidable. Our second goal is to develop an argumentation model that acknowledges the difficulties that machine learning will encounter over complex domains and will still be able to allow agents interacting in these domains to reach mutual intelligibility.

Coverage Preservation Our model uses inductive learning to classify a domain. This means that a set of generalization is found for each class of the domain, where each example of the class has an *is-a* relation with one of the generalization of the corresponding set. This means that if our learning is not perfectly accurate, some examples might be not related to any generalizations. These examples become *uncovered.* Our third goal is to develop an argumentation model that does not cause covered examples to become, after argumentation, no longer covered by some concept. When an agent changes its classification by learning a new set of generalizations, the agents should make sure that the new classifications cover at least the same examples than the previous one. Ideally, the final classification of our agents, after mutual intelligibility is achieved, should encompass more examples than the initial one.

Efficiency The agents have access to different subsets of their overall context. While these two subsets can overlap, it should not be taken for granted that our agents know anything about each other's context. We can we measure the ratio of the overlap between the two local contexts relative to the overall context. Our fourth goal is to keep the information exchange low, even in the worst cases where the overlap is low or zero.

Simplicity Our agents classify their domains in different concepts. Since disagreement on the meaning of concepts can be interpreted as differences in the classification of a domain, reaching a mutual intelligibility requires from our agents to change the classification of their domain. Since the domain is classified into different domains, changing the classification of the domain is tantamount to creating new concepts. Our goal is to allow the agents to change their classifications in order to reach mutual intelligibility by creating new concepts but not creating unnecessary concepts. Our fifth goal is to limit the number of concepts that are incorporated in the final classification of our agents once mutual intelligibility has been reached.

Scalability Our sixth goal is to develop an argumentation model of argumentation where the amount of information exchanged by the agents does not hugely increases with the size of their domain. If the information exchanged by the agents would increase disproportionately with the size of the domain, there would be a theoretical size of domain for which the agents would exchange as much information as there actually is in the domain. Since our fourth goal is to limit the exchange of information between the agents, it is important to verify that this exchange stays limited even when the amount of information not shared by the agents is significant.

1.3.3 Contributions

By developing an approach that combines symbolic concept learning and multiagent systems, extended with a semiotic approach which allows example-sign revisions during communication, our contribution is related to three main fields in artificial intelligence. Concerning the symbolic concept learning, we contribute by removing the common assumption that examples have to be definitively associated to the sign that labels them. Concerning agent-based argumentation, we contribute by providing a model of argumentation for agents that can handle semantic heterogeneity and allows two agents to achieve mutual intelligibility. Concerning semantic alignment, we propose both a model and an approach that do not need fixed meaning in order to allow mutual intelligibility between the agents, and can address semantic heterogeneity problems in runtime. Therefore, this reduces the need of a previous phase that tries to resolve all semantic heterogeneity problems before the runtime of open multiagent systems.

By developing a model that can satisfy the hypotheses listed in the previous section, we contribute to the field of multiagent systems with a model that is able to work on different situations of semantic heterogeneity and performs well in effectively reaching mutual intelligibility, while having a relatively low cost in terms of information exchanged within the system.

1.4 The Thesis

This monograph begins with a presentation of the related work in Chapter 2, starting with an overview of the ESSENCE Network, in which this thesis was commenced and integrated. We present the relation of our work in the field of machine learning symbolic learning, and give an overview of the coordinated inductive learning, an approach to symbolic learning on which we draw similarities with our thesis. The relevance of the ontology alignment field to the creation of our map of relations between concepts is also presented, along with researches in ontology alignment that also consider a dynamic and interactive approach to concept mapping. Finally, we discuss the field of computational semantics, another field in artificial intelligence that uses semiotic elements to improve the knowledge representations of intelligent systems.

After the presentation of the related work, we give an in-depth description of the problematic issues involved in reaching mutual intelligibility in situations of semantic heterogeneity in Chapter 3. We address this problematic under the general angle of classification in multiagent systems, and give elements of notation to define the goals of our model.

The presentation of our approach is in Chapter 4, where we explain how to dealing with a lack of mutual intelligibility between two agents. Chapter 4 presents the general elements of our model. This chapter uses a notation inspired by semiotics. The field of semiotics deals with the relation of signs and their meaning, and the semiotic elements defined in Chapter 4 are useful during the argumentation over concept meaning, as they allow the agents to make explicit the relationships between their vocabulary and the partitions that they make of their contextual domain. Furthermore, Chapter formalizes the notion of pairing relation between concepts, allowing the expression of mutual intelligibility as a specific state of pairing relations between the concepts of two contrast sets. The formal definition of mutual intelligibility is followed by a formal definition of the notion of disagreement. These definitions are expressed as a combination of pairing relations between pairs of concepts and their signs. Moreover, we define a typology of these disagreements that facilitates later the phases of agreement identification and agreement resolution, that are at the core of our model. Chapter 4 introduces the notion of r-triplets, that contains the relevant information about pairs of concepts to infer their pairing relations. We also present how, by exchanging r-triplets, the agents can infer the pairing relation of their concepts in the overall context.

Our argumentation model is presented in Chapter 5. We start with the description of the different capabilities that our agents can use to interact, and how these capabilities are used by the agents to reach mutual intelligibility. We also explain how agents can *create new concepts*, when needed to resolve complicated disagreements, using a specific argumentation model in order to reach mutual intelligibility. This model, presented in assumes that the elements of symbolic learning used in our model can learn concepts that classify with a perfect accuracy (we call this the error-free model).

Chapter 6 introduces the notion of *degree of error*, that reflects the possibility for symbolic learning algorithms to classify with some error degree. We explain the impact of classification error in on our approach, presented in Chapter 4, and in different parts of our model, presented in Chapter 5, that assume error-free learning. Chapter 6 then introduces the enhancements needed to obtain an error-tolerant model, able to reach mutual intelligibility.

Chapter 7 and Chapter 8 detail the two different strategies of our argumentation model that our agents can adopt. Chapter 7 details the systematic strategy, a strategy in which the agents, immediately after meeting, engage in resolving their disagreements over the meaning of their concepts. Chapter 8 details the lazy strategy, where the agents play a naming game and resolve disagreements over the meaning on-demand, when a disagreement occurs during the the naming game.

These two strategies are then exemplified in Section 9, in order to the details of the strategies when applied to specific disagreements in a specific domain, and explaining the steps taken from disagreement identification to resolution. The exemplification of these problems and their process of resolution also clarifies how each of our two strategies workand also highlight the differences and similarities between the two strategies.

Chapter 10 presents an experimental evaluation of both strategies. In this chapter, the six hypotheses about our model presented in Section 1.3.2 are detailed and experimentally evaluated, both on the systematic and lazy strategies. First and foremost, the experiments shows the *efficacy* of our approach, in both strategies, since disagreements of all types (and their combinations thereof) are resolved and mutual intelligibility is achieved.

Moreover,the experiments prove that both strategies can allow agents to achieve mutual intelligibility over in different of application domains. The results also show that the mutual intelligibility is not reached at the cost of an extensive exchange of information between the agents. Chapter 10 also investigates how the lazy approach can be preferred to the systematic approach when the agents have a large set of concepts but only use a few of them during their interactions. The monograph is concluded with Chapter 11, that offers a discussion over the contributions of our argumentation model and the empirical results achieved. The chapter closes with a discussion of the future research work on semantic heterogeneity that can follow.

This monograph has two appendices. Appendix A offers a general presentation of the three parameters of our model. These three parameters are used as independent variables in simple experiments, in order to justify our choices of parameters in Chapter 10. Appendix B gives a list of all the types of messages that can be used in our model along with their and the type of elements that they can carry.

Chapter 2

Related Work

2.1 Introduction

This chapter describes the relation of our work with several fields of artificial intelligence: symbolic concept learning, agent-based argumentation, ontology alignment and computational semiotics. However, since the topic of this Ph.D was engendered inside an International Training Network focused on the "Evolution of Shared Semantics in Computational Environments", we will start by describing the ESSENCE Network and its relationship with the research presented in this monograph.

2.2 The ESSENCE Network

ESSENCE was a European research training network that conducted world-leading research into the evolution and negotiation of meaning among human and artificial agents. ESSENCE investigated semantic technologies, language games, multiagent communication, ontology learning, and human dialogue, which all contribute to a broader research vision of diversity-aware AI. This vision emphasises creating next-generation AI technologies that can be used to bridge the gap between heterogeneous agents by exploring how representation, reasoning, and interaction can be used to allow diverse collectives of agents to share information and knowledge, coordinate their activities, and combine their individual capabilities.

The work of ESSENCE was divided in four thematic areas, each addressing core research questions: Rational Action and Communication, Data and Communication, Human Communication and Representation and Reasoning. This thesis is part of the work done by the group working on Representation and Reasoning. The other member of this group was Paula Chocron, who researched solutions to semantic heterogeneity for ontology alignment in the context of two agents having similar concepts but no shared correspondences between their meanings, nor assumption on a particular set of properties over the examples considered as shared knowledge (Chocron and Schorlemmer, 2017). Instead, the agents use expectations from their interactions to align each other's concepts. A given example involves one person ordering a beer in a foreign country, and obtaining two interrogative words as an answer. While this person does not speak the language to which these words belong, this person can use its prior knowledge on ordering beers to understand that these two words probably corresponds to the main colors for beers, blonde or dark. Continuing the interaction, this person becomes more on more certain about the meaning of each word used by the foreign barman.

On the contrary, our approach focuses on agents that already share a common description language, but struggle to identify which specific relations of their described examples with a dynamic set of signs can reduce errors in communication. Our approaches does not assume prior knowledge over a certain set of expectations that they should respect in their interactions other than the protocol they use to argue over the meaning of their concepts.

2.3 Relation to Symbolic Concept Learning

The problem of symbolic concept learning, formalized as the creation of a generalization of a class from a set of examples and given a classification of these examples divided in positive examples (examples) and negative examples (counter-examples), has been expressed as a search problem (Mitchell, 1982). Therefore, a multiagent system regrouping a large set of examples should be able to have different agents searching different parts of the space of generalizations, starting from the generalization of their own examples, and communicating to coordinate the learning process while one agent can prevent another agent to search in sub-spaces that it has already explored unsuccessfully.

Therefore, the idea to use multiagent learning has been extensively studied from different perspectives (Stone and Veloso, 2000). The predominant approach in multiagent settings has been the reinforcement machine learning (Littman, 1994). Our work, however, is more related to classification in MAS (Modi and Shen, 2001). Moreover, our approach is collaborative instead of an adversarial (Stone and Veloso, 1998), and we focus on learning from explicit rather than implicit communication (Aras et al., 2004). Research work in Case-Based Reasoning is also based on the idea of agents being able to exchange cases – which we call examples – in order to retrieve helpful cases from other agents in order to solve new solutions (Prasad et al., 1996). Our approach, however, is more similar to AMAIL (Ontañón and Plaza, 2015) in the sense that it does not only apply to case retrieval. Moreover, the items corresponding to cases in our approach are examples and the solutions are arbitrary symbols, which brings an additional difficulty to the argumentation.

Our approach is using symbolic inductive learning of concepts, which is a type of machine learning that has already been used in multiagent learning. The MALE (Sian, 1991) and the DRL (Provost and Hennessy, 1996) algorithms are both inductive-learning algorithms that rely on a multiagent system to build a set of rules that applies to all their distributed knowledge. However, each of these algorithms works as a single algorithm working on all data, parallelized among different agents. Therefore, these algorithms assume that the different parts of the data-set distributed among the agents are coherent, and semantic heterogeneity cannot addressed in these approaches.

2.4 Relation to Agent-based Argumentation

Argumentation has been presented as a promising approach to create generalizations on inconsistent knowledge, based on the construction and the comparison of arguments. Argumentation on MAS is in fact a topic that has been broadly studied in terms of logics, protocols and languages that support argumentation, argument selection and argument interpretation (Rahwan and Simari, 2009)(Gómez and Chesnevar, 2003). These arguments constrain the search among possible hypotheses between agents, and can also direct the search towards hypotheses that are more comprehensible in the light of expert's background knowledge (Možina et al., 2007). For instance, some approaches specific to case-base reasoning allow agents to gauge the strengths and weaknesses of other agents, such that the agents retain only certain cases provided by other agents that are able to improve their individual performance (Ontañón and Plaza, 2007a).

Advantages of argumentation-based approach includes, for the purpose of classification, identifying the reasons that led to the classification, classifying examples even when the set of training examples is inconsistent, and considering more general preference relations between hypotheses while the results obtained from centralized approach of symbolic can be retrieved (Amgoud and Serrurier, 2007).

Previous systems have used voting methods or theory refinement techniques to integrate "local concepts", concepts that are only valid on a portion of the agents' overall context. These techniques carry the risk of bypassed parts of the hypothesis that are correct. Argumentation models always have to be careful to not bypass those parts and propose specific strategies to overcome this issue (Davies and Edwards, 1996).

The closest argumentation-based model to ours is AMAIL (Ontañón and Plaza, 2015), argumentation approach for agents to argue about concepts learnt by induction. AMAIL proposes a model of argumentation where multiple agents learn separately some concepts by inductive learning, over N different sets of examples. Then, the agents propose an intensional definition for each concept, and argue over each other's intensional definitions, they exchange arguments to improve their local and/or global accuracy until they have no more useful information to exchange: no more g-arguments can be generated, because their classifications are very similar. In this approach, the intensional definitions made by the agents start being different but gradually cover more and more common examples, until the agents achieve a very similar classification. While this is not the only approach that uses incremental changes to create equivalent but not equal concepts (Bourgne et al., 2007, 2010), the model of AMAIL can use arguments that are not only examples or counter-examples, but also generalizations over examples, which decreases the volume of information that needs to be exchanged in order to share the knowledge of the agents. An arguments that uses generalizations attacks another argument when the first is subsumed by the second but predicts a different class. The attack success or failure depend on the relative support given by the number of examples covered by those generalizations.

In order to generate such arguments and counter arguments, AMAIL uses the ABUI algorithm. The ABUI algorithm is an inductive algorithm that, for a given set of positive and negative examples, a set of accepted arguments and an argument to defeat, generates a counter-argument against the argument to defeat.

Our approach differs with AMAIL on the assumption that example-signs associations are fixed when received. This also defines the difference in domain of experiments between the two approaches. While AMAIL focuses on agents learning over different partition of a same context, the agents of our approach do not necessarily work with local contexts from the agents that could be regrouped without having two examples labelled differently. However, the ABUI algorithm is a central element of a disagreement resolution strategy, namely the creation of new concepts through argumentation and we will use it in our research for inductive generalization and for argument generation.

Another component of the AMAIL platform that is used in our model is a similarity measure based on anti-unification (Ontañón and Plaza, 2012). The anti-unification similarity measure, used in refinement graphs and inductive learning, determines how close two generalizations or examples are from each others by counting the steps necessary to reach their anti-unification in the generalization space. In the context of our approach where signs and examples are not associated, the anti-unification distance can be useful to classify an example that is not covered by any concept.

While the AMAIL approach is the closest from ours, two other approaches to argumentation-based concept creation in MAS are the argumentation frameworks AMAL (Ontañón and Plaza, 2007b) and PADUA (Wardeh et al., 2009). While the AMAL framework also focuses on the idea of learning from argumentation, the goal of AMAL is to argue on the classification of certain examples and does not concern rules learned through inductive learning. The PADUA framework is an argumentation that allows two agents to discuss association rules. This framework is interesting, as it gives precedence to the association rules of the agents over the initial class of the examples. In the terms used in this thesis, the left-path associations are also favored over the right-path associations. However, the agents of PADUA work with both strict and defeasible rules, while our agents do not rely on strict rules at all. Another interesting framework is SMILE (Bourgne et al., 2007, 2010), which is somewhat similar to AMAL but only allows the exchange of examples, and not the exchange of rules.

2.5 Relation to Ontology Alignment

Mapping and refining concepts is also a domain of interest of Ontology Engineering. In particular, changing the underlying semantics of an ontology by allowing artificial intelligence systems to manipulate their own internal representations automatically has been considered of a great significance for artificial intelligence (Bundy and McNeill, 2006).

If we focus on the use of ontologies concerning agent systems, some research has been done in order to repair ontology alignments that appear to be inaccurate by using contextual interactions between agents (Chocron and Schorlemmer, 2017; Euzenat, 2017). These approaches focus on systems where each agent has a different ontology that cannot be accessed by other agents. Correspondences are then found by assuming alignment, testing the assumption with the classification of an example by both agents, and revising the alignment according to the results of the classification. On the contrary, our model assumes a certain degree of correspondence between the agents ontologies, in particular on the matter of the concepts used in the description of examples. Moreover, concept alignment is not an end in itself, but a step in the creation of a collection of concepts shared by the agents (including the creation of new concepts). Trojahn & al. have extensively investigated the utility of different variations of value-based argumentation frameworks (Isaac et al., 2008; Trojahn et al., 2008a,b, 2012) in order to match ontologies. Their research focuses on how preferences over values in different audiences give different acceptability degrees to arguments, and how this can impact the outcome of argumentation. In our model, the agents do not take into account audiences or values since their goal is to develop a shared semantic field between the two agents, even creating new concepts when necessary to surmount specific disagreements.

Another recurrent problem addressed in formal ontologies is the fact that in the case of data created by heterogeneous sources, those sources will use specific terminology over their own data, meaning that data from one source will be incompatible with data from another source. Ontology therefore shares the problem of extracting meaningful information from large data sets with coordinated machine learning, adding an element of semantic heterogeneity. A solution to this problem is to access dynamically the information (Halpin and McNeill, 2013), meeting here the idea of contextual meaning. In the case of agents having two different ontologies, this translates into aligning dynamically the concepts from their ontologies without requiring full access to the ontologies of other agents and works entirely automatically and dynamically (McNeill and Bundy, 2007).

In this perspective, argumentation has also be seen as a solution to semantic heterogeneity with agents using ontologies being able to change their choices of vocabulary used to represent concepts through the creation or exchange of generated arguments, that support or reject possible correspondences (Laera et al., 2007). In order to evaluate this support, measures of agreement and disagreement based on the search of logical inconsistencies have been proposed (d'Aquin, 2009). This measure of agreement and disagreement differs from our approach, where disagreements are qualitative and therefore not measured but counted, with respect to some expected degree of error.

On a more general note, cognitive science is helpful to understand the duality of our approach and the importance of machine learning and formal ontologies in the two resulting elements of our approach. Goldstone et al. (Goldstone and Rogosky, 2002), for instance, differentiates between "external grounding" theories of meaning (where concepts depends on their connections to the external world), and "conceptual web" theories of meaning (where a concept's meaning depends on its relations to other concepts within the same system). While the fact that the former theory relates more to machine learning and the latter to formal ontologies is far from being a universal truth, this differentiation illustrates how concepts modeled as semiotic triangles relates to an external grounding approach more present in machine learning, and how grouping these concepts in contrast sets and making them interdependent with pairing relations introduce notions usually found in ontologies.

2.6 Relation to Computational Semiotics

Computational semiotics is a field that focus on describing how notions of semiotics, which is a field of human sciences, interconnects with the study of intelligent systems. In particular, computational semiotics aims to provide a set of methodologies that use concepts and terminology of semiotics to design frameworks suitable for artificial agents (Gudwin and Gomide, 1997; Gudwin, 1999). Computational semiotics, as our approach also does, focus on the necessary representations that intelligent agents need in order to understand their own language, and propose specific approaches to their modelling (Doeben-Henisch, 2009; Guerrero et al., 1999; Rieger, 1997). Semiotic linguistics, however, generally put more emphasis on agents using natural languages (Rieger, 1997).

2.7 Conclusion

Our approach relates mainly to the intersection of three fields of artificial intelligence, namely symbolic concept learning, agent-based argumentation and ontology engineering. More specifically, the work in the field of cooperative learning agents is the more relevant to our thesis, as shown in the reuse of some algorithms from AMAIL (Ontañón and Plaza, 2015) in our implementation. Loosening the relation between signs and examples in our approach, however, places us in a really different paradigm where initial example-sign relations cannot be accounted as a universal truth by our agents in order to build and argue about their concepts. Relations between concepts therefore become as relevant as the extensions of these same concepts in the modification and creation of concepts, an element that is particularly present in the field of ontology alignment. The scope of our approach meets the current necessity in ontology alignment to develop new models to dynamically and contextually change concepts' meanings. However, our main goal stays mainly focused on the cooperative learning goal of creating new meanings through agent interactions. While our approach focuses on the computational resolution of disagreements occurring during coordinated learning, the semiotic dimension of our model that mirrors some aspects of human concept representations also makes our work related to computational semiotics.

Chapter 3

Classification in Multi-Agent Systems

3.1 Main Problem

The issue addressed in the present thesis is not argumentation on meaning *per se*, an argumentation being always a mean to attain a goal. And in this thesis, the goal that needs to be reached — through an argumentation — is the attainment of a contextual *mutual intelligibility.* A good illustration of what a contextual mutual intelligibility means can be given through the example of a naming game between two agents. If two agents receive a same set of examples, that we will call their context U, we want those two agents to be able to agree, for each example e that belongs to U, on a single sign s to name e. If these agents are able to perform this task, we say that they have reach a contextual mutual intelligibility over the context U.

Therefore, the goal of our argumentation process is to provide two agents with the capability to classify in a mutually consistent way a set of examples (that we call a *context* U), and by mutually consistency we mean that both agents classify each example in U with a same label (that we will call a *sign*). Since we said that these agents *agree* on a certain sign for each example, we also call this situation of mutual intelligibility an *overall* agreement on the example-sign associations in the context U. On the contrary, if the two agents associate a given example e with two different signs, we say that the agents *disagree* on their example-sign associations. In this chapter, we will present an approach for learning agents to play the naming game in general terms, before introducing the more specific lexicon to our approach in Chapter 4. We are going to present the notion of mutual intelligibility in terms of example signs associations, and define the space of experiments that our argumentation model can tackle.

3.2 Notation

3.2.1 Sets of associations

In the naming game, the classifier agents are presented one example e at a time and each agent associates this example e with a sign. If both agents have associated the example e with the same sign, the agents have agreed on the sign of e and scored a success. Therefore, the basic action that an agent should be able to do in order to play the naming game is to associate an example with a sign.

Definition 1 (Example-sign association). *The association between an example e and a sign s by an agent A_k is written $e\ {}_{Ak}{\mapsto} s$.*

We can generalize the notion of association to not only one example, but an entire a set of examples U, that we will call a *context*. An agent A_k associates signs from a *lexicon* (a set of signs) S to the examples of the context U. This results in a set of example-sign associations.

Definition 2 (Set of associations). *A set of associations between the examples of $U = \{e_1, \ldots, e_n\}$ and the signs of $S = \{s_1, \ldots, s_m\}$ is written as: $U\ {}_k{\mapsto} S = \{e_1 \mapsto s_i, \ldots, e_n \mapsto s_j\}$.*

Classes

Example-sign associations can be grouped in *classes*. Classes are sets of examples that are related by their signs among a specific set of example-sign associations. Usually, classes group examples that are associated with the same sign.

Definition 3 (Class). *A class $U(\mapsto s)$ is a subset of examples from U such that $U(\mapsto s) = \{e \in U | e \mapsto s\}$. A consequence of this is that the agents cannot associate zero or more than one sign(s) to an example from U.*

However, there are more complex situation where some examples are associated with multiple signs. A set of examples that are associated with a unique *set* of signs can be grouped in what we call a *polylexematic* class.

Definition 4 (Polylexematic class). *A polylexematic class $U(\mapsto \{s_1, \ldots, s_n\})$ is a subset of examples from U such that $U(\mapsto s) = \{e \in U | e \mapsto s_1 \vee \ldots \vee e \mapsto s_n\}$.*

In our presentation of mutual intelligibility, we explained that the agents should be able to associate a same sign –singular– to a same example. Since the polylexematic classes are associating more than one sign to an example, they cannot allow the agents to reach mutual intelligibility by our terms. Since we presented the mutual intelligibility in term of agreements, we can relate polylexematic to a factor of disagreements in our approach.

Properties of sets of associations

Before continuing with the presentation of our model, we present some useful properties of sets of associations that will help us to define both the criteria that are required from learning agents in order to consider that they have reached mutual intelligibility, and the space of the experiments that our model can tackle.

Consistency An important notion linked to the sets of associations in the notion of consistency. A set of associations $U \mapsto S$ is said to be consistent if it maps each example from U to exactly one sign from S. In term of classes, it means that for any pair of classes $U(\mapsto s_i)$ and $U(\mapsto s_j)$ in $U \mapsto S$ we have $U(\mapsto s_i) \cap U(\mapsto s_j) = \emptyset$ if and only if $U \mapsto S$ is consistent.

Property 1 (Consistency). *A set of associations $U \mapsto S$ is consistent if and only if, for each example $e \in U$, there is no pair of associations $e \mapsto s$, $e \mapsto s'$ in $U \mapsto S$ such that $s \neq s'$.*

Property 2 (Classes in Consistent Association Sets). *In a set of associations $U \mapsto S$, each pair of classes $U(\mapsto s_i)$, $U(\mapsto s_j)$ in $U \mapsto S$ verify that $U(\mapsto s_i) \cap U(\mapsto s_j) = \emptyset$ if and only if $U \mapsto S$ is consistent.*

Property 3 (Polylexematic Classes in Consistent Sets). *If a set of associations $U \mapsto S$ is consistent there are no polylexematic classes in $U \mapsto S$.*

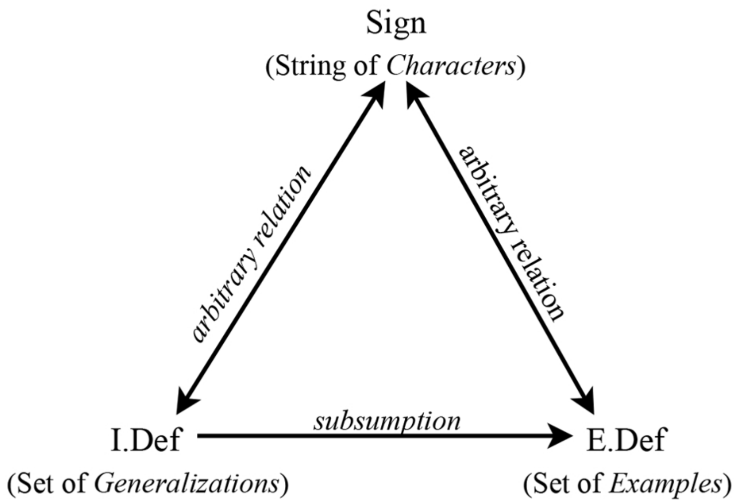


Figure 3.1: Representation of the notion of concepts in our model as a semiotic triangle.

3.2.2 Example-Sign Associations from Learning Agents

The central element of the naming game and by extension of mutual intelligibility are the example-sign associations. While there are multiple ways to associate signs with examples, the agents from this thesis are learning agents, able to learn concepts through supervised learning and to use these concepts to classify examples. Our learning agents have two main strategies to make example-sign associations, and we will now present the notation for these strategies in order to make explicit which strategies the agents use, and which problem each strategy can pose to mutual intelligibility.

In supervised learning, an agent A_k receives a *consistent* set of example-sign associations $U_{o}{\mapsto}S$ from the experimenter. These associations are memorized by A_k in a set of example-sign associations $U_{k}{\mapsto}S$. For the moment, the examples U are the only examples the agent A_k has knowledge of, and are called the *local* context of the agent. Comparably, the signs S that an agent has knowledge over is the *local* lexicon of this agent. Prior to having any supervised learning, the agent can already make example sign associations over the examples U by looking up its set of example-sign associations $U_{k}{\mapsto}S$ which constitutes an index of received example-sign associations. However, A_k cannot make example-sign associations for new examples.

The supervised learning we are interested in is *symbolic concept learning using inductive techniques* that create generalization from the training examples. Supervised learning takes place by using each example e from a class $U(\mapsto s)$ as an input and the sign s as an expected output. Once the supervised learning is done, the agent should have learned a set of generalizations such that, provided any example e and any generalization g, any agent could say if e can be associated to g. Each of the generalizations that are learned are associated to the sign s. In this thesis, the agents are inductive learners that use the feature-term formalism presented in Section 4.1.1 to represent examples and generalizations. The association between an example e and a generalization g can be tested through the relation of subsumption $g \sqsubseteq e$. After the learning of a set of generalizations $I = g_1, \ldots, g_n$ over a set of example signs $E = U(\mapsto s)$, the agents have learned a new *concept* C. A concept regroup three elements: the sign s, the set of generalizations I, and the set of examples E. The set of generalizations of C is called the *intensional* definition of C, while the set of examples of E is called the *extensional* definition of the concept C. These three elements, called the semiotic elements, will be defined more thoughtfully in Chapter 4. We can envision the three semiotic elements of a concept in a semiotic *triangle*(Ogden and Richards, 1923) similar to Figure 3.1, that is a representation of a concept in our approach. For any giver concept C, we use the following notation: $s(C)$ is its sign, $E(C)$ is its extensional definition, and $I(C)$ is its intensional

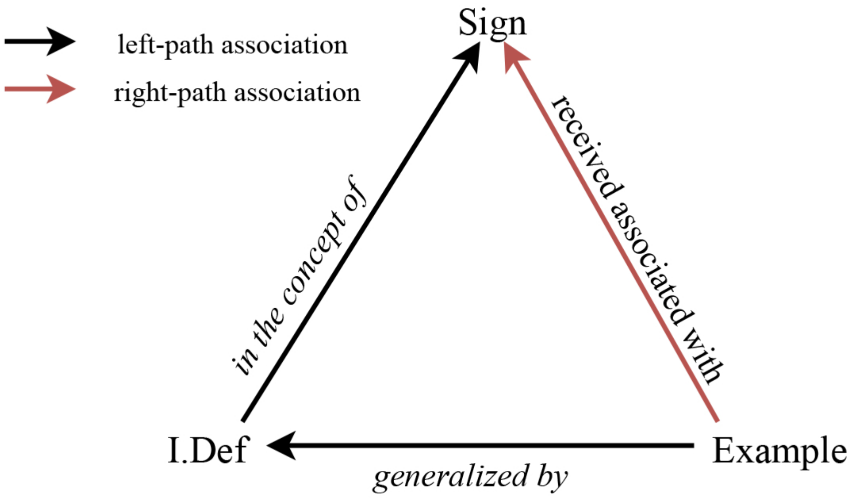


Figure 3.2: Paths of example-sign associations offered to the agents. The right path is considered as the "objective truth" as it has been received by the experimenter, while the left path is an inference made through learning that can be used in a more diverse set of scenarios to make new associations.

definition.

After having learned a set of concepts $C_1, \ldots, C_n$ learned through supervised learning, the agent A_k can associate any example e with a sign s from the lexicon $s(C_1), \ldots, s(C_n)$ by using the generalizations from $I(C_1), \ldots, I(C_n)$. First, the agents looks at which generalization $g \in \bigcup_{i=1}^{n} I(C_i)$ subsumes the example e. Then, the agent looks at which concept C the generalization g is belonging to, and associates e with $s(C)$. A_k ends up with a new example-sign association $e \mapsto s(C)$. Multiple generalizations can subsume a same example e, and these generalizations can be from different concepts $C_a, \ldots, C_n$. In that case, the agent creates n new associations $e \mapsto s(C_a), \ldots, e \mapsto s(C_n)$.

In this section, we described two strategies, for a learning agent A_k that has received a set of associations $U\,{}_{k}{\mapsto}S$, to associate an example e with a sign:

- looking into $U\,{}_{k}{\mapsto}S$ for an already existing example-sign association involving e.
- looking into A_k's set of generalizations to find a generalization g that subsumes e, and use the sign of g's concept.

We call these two way of associating example with signs the *left*-path and the *right*-path associations, in reference to the path that they follow in our model of a concept represented as a semiotic triangle. The two paths are shown in Figure 3.2. We incorporate this distinction in the notation of example-sign association by adding the letter l or r for left path or right path the the usual notation $e\,{}_{k}{\mapsto}s$. If the example e is associated to the sign s through the left path by agent A_k, we write $e\,{}_{k}^{l}{\mapsto}s$, while, if the example e is associated to the sign s through the right path by agent A_k, we write $e\,{}_{k}^{r}{\mapsto}s$. The consistent set received for the learning phase of training by A_k is always written $U\,{}_{o}{\mapsto}S$. An accurate supervised learning from the agent A_k results in:

$$U\,{}_{k}^{l}{\mapsto}S = U\,{}_{k}^{r}{\mapsto}S = U\,{}_{o}{\mapsto}S.$$

3.2.3 Selecting Paths in Naming Games

The agents have two methods, expressed as left and right paths, to associate one example e to a sign s in the naming game. We already mentioned that a perfectly accurate supervised learning process would result in equivalent associations made by both paths. However, perfectly accurate learning process is not always possible. In this situation, an agent A_k that is presented an example e during the naming game, has to decide whether to use $e \overset{r}{\underset{k}{\mapsto}} s$ or $e \overset{l}{\underset{k}{\mapsto}} s'$ if the signs s and s' are different.

The agents always use their left-path associations to play the naming game for a simple reason. An example e that is presented to the agents in the naming game might not be in any of the examples U_k that the agent A_k has received from the experimenter. In this case, the agent A_k cannot use its right-path associations to name e. Since only one path should be chosen to name examples, the left path is therefore privileged over the right path. This has a direct consequence on our model: if the agents encounter disagreements during the naming game, it is their left-path associations that should be changed in order to allow the agents to reach mutual intelligibility. While the agents also change their right path associations, this is only to make the right path consistent with the left path but this is not a necessary step in our model.

3.3 Disagreements in the Naming Game

There can be two factors that cause disagreements during a two learning agents naming game. The first is the presence of differences between the set of example-sign associations received by the agents. The second is the presence of errors during the agents' learning of concepts. The objective of our thesis is two explore a model that can achieve mutual intelligibility in scenarios that combine both factors.

3.3.1 Differences in Received Sets of Associations

In a naming game involving two learning agents, an experimenter can give different sets of example-signs associations to the agents. From now on, we will consider that the general case is indeed an agent A_1 receiving a set of associations $U_1 \underset{o}{\mapsto} S_1$ and a second agent A_2 receiving a set of associations $U_2 \underset{o}{\mapsto} S_2$ from the experimenter, such that:

$$U_1 \underset{o}{\mapsto} S_1 \neq U_2 \underset{o}{\mapsto} S_2.$$

The first factor that can cause disagreements during the naming game is the inconsistency of $U_1 \underset{o}{\mapsto} S_1 \cup U_2 \underset{o}{\mapsto} S_2$. In this case, the differences between $U_1 \underset{o}{\mapsto} S_1$ and $U_2 \underset{o}{\mapsto} S_2$ will be responsible of disagreements during a naming game. Let e be an example that belongs to both U_1 and U_2, but is associated to s in $U_1 \underset{o}{\mapsto} S_1$ while it is associated to s' in $U_2 \underset{o}{\mapsto} S_2$. If the agents are using their right path associations to name e, the agents A_1 will use s to name e in the naming game while the agent A_2 will use s'. Moreover, we mentioned in the previous section that an accurate supervised learning from the agents results in:

$$U \overset{l}{\underset{1}{\mapsto}} S = U \overset{r}{\underset{1}{\mapsto}} S \text{ and } U \overset{l}{\underset{2}{\mapsto}} S = U \overset{r}{\underset{2}{\mapsto}} S.$$

Therefore, even if the agents use their left-path associations, they are expected to have a disagreement over the example sign association of e.

3.3.2 Differences in Learning

The agents can make different types of errors when they learn concepts. The two types of error that we will now detail are different from the first and second type error often encountered in classification. Let A_k be an agent that is tasked to learn a concept C which corresponds to a class $U(\mapsto s)$, from the set of associations $U \mapsto S$. The first type of error that A_k can do, is to create C as a classification of $U(\mapsto s)$ with errors. In this case, generalizations of $I(C)$ either subsume some examples that are not in $U(\mapsto s)$ or do not subsume some examples that are in $U(\mapsto s)$. This type of error is common in supervised learning, but this is not the type of error we are interested in at the moment (but we address it later in Chapter 6).

The second type of error that the agent A_k can make is to learn concepts are two specific. Let U, U_1 and U_2 be three contexts, s_1 and s_2 be two signs, and S be a lexicon such that $S = \{s_1, s_2\}$. Let $U \mapsto S$, $U_1 \mapsto S$ and $U_2 \mapsto S$ be three consistent sets of example-sign associations such that:

- $U \mapsto S = U_1 \mapsto S \cup U_2 \mapsto S$,
- $U_1 \mapsto S \cap U_2 \mapsto S = \emptyset$.

The agent A_k is this time tasked to learn a concept C that generalizes the class $U_1(\mapsto s_1)$, using only the sets of associations $U_1(\mapsto S)$ to learn C. This time, the agent A_k achieve a successful learning in the context of U_1 and can correctly classify all the examples from $U_1(\mapsto s_1)$. In a second time, A_k is tasked to use its concept C in the context U_2 to classify the examples from $U_2 \mapsto s_1$. The classification of $U_2 \mapsto s_1$ by C is unsuccessful, the concept C either generalizing some examples from $U_2(\mapsto s_2)$ or not generalizing some examples from $U_2(\mapsto s_1)$. The agents A_k has still successfully learn the classification that A_k was tasked with, but A_k cannot transfer the knowledge that it learned with C to other contexts than U_1. This type of error is called *overgeneralizing* when C fails to generalise examples from $U_2(\mapsto s_1)$, and *undergeneralizing* when C generalises examples from $U_2(\mapsto s_2)$. This type of error is also common in supervised learning. *Under/overgeneralizing* is the type of error we are now interested in.

In Section 3.3.1, we saw that two agents could be sent different sets of associations to learn on, and that this would result in disagreements during the naming game. The two sets off associations sent in Section 3.3.1 had an union that was inconsistent, resulting in disagreements during the naming game. In the present case, however, we can imagine two agents A_1 and A_2, A_1 receiving a set of associations $U_1\,{}_o{\mapsto}S$ and agent A_2 receiving a set of associations $U_2\,{}_o{\mapsto}S$ from the experimenter, such that $U_1\,{}_o{\mapsto}S \cup U_2\,{}_o{\mapsto}S$ is consistent. Let e and e' be two examples such that:

- $e \mapsto s \in U_1\,{}_o{\mapsto}S$ and $e \mapsto s \in U_2\,{}_o{\mapsto}S$, and
- $e' \mapsto s \in U_1\,{}_o{\mapsto}S$ and $e' \notin U_2$.

In this situation, if the agent A_2 has learned overgeneralized concepts that still classify correctly the classes of $U_2\,{}_o{\mapsto}S$, then A_2 will associate the example e with the same sign s as the agent A_1 during the naming game. However, as the example e' is not in U_2, the agent A_2 might associate e' to a different sign than s during the naming game, or even no sign at all. The agent A_k, if it has learned correctly to classify the classes of $U_1\,{}_o{\mapsto}S$, will certainly associate e with s.

In supervised learning, whether or not an agent will under-generalize or overgeneralize its concepts depends on the sets of associations used to learn those concepts with regard to the sets of associations used to evaluate the learning of those concepts. We can therefore see the proportion of two agents learning concepts on a context U_k to be undergeneralizing or overgeneralizing in a more general context U_O partially as a property of U_k and U_O, not only as a property of the agents. We cannot measure how the degree of under/overgeneralizing of the concepts of two agents before knowing the context in which the naming game is taking place. We can consider that, by default, the naming game takes place in an *overall* context $U_1 \cup U_2$ that encompasses the examples known by at least one agent —i.e. the union of both sets of examples. In this situation, we can introduce the notion of homogeneity to represent how well a context allows two agents to not undergeneralize or overgeneralize their concepts.

Definition 5 (Homogeneity)**.** *Let A_1 and A_2 be two agents. A set of example-sign associations $U \mapsto S$ is homogeneous if, for any pair of its subsets $U_1 \mapsto S$ and $U_2 \mapsto S$ such that $(U_1 \mapsto S) \cup (U_2 \mapsto S)$, we have $U_k \;{}_{A1}\!\!\overset{l}{\mapsto}\! S = U_k \;{}_{A2}\!\!\overset{l}{\mapsto}\! S$.*

As we mentioned, the inductive learning of a concept can be not perfectly accurate. This is why we introduce the notion of a degree of homogeneity.

Definition 6 (Degree of homogeneity)**.** *Let A_1 and A_2 be two agents. The degree of homogeneity d_h of two sets of example-sign associations $U_1 \mapsto S$ and $U_2 \mapsto S$ is given by the formula:*

$$d_h = \frac{1}{2} \sum_{k=1}^{2} \frac{|U_k \;{}_{A1}\!\!\overset{l}{\mapsto}\! S \cap U_k \;{}_{A2}\!\!\overset{l}{\mapsto}\! S|}{|U_k \mapsto S|}$$

3.4 Experimental Domain of our Model

We presented the notion of mutual intelligibility in terms of agreements in naming games, and we described how two learning agents could play a naming game. We also presented how disagreements can arise in naming games when two agents are learning their concepts. We will now define the space of the experiments in which the agents will encounter disagreements during their naming game, and for which our model will provide solutions to reach mutual intelligibility. In order to do so, we will endorse the role of an experimenter that sets up these experiments, and present the set of actions that we can take in order to create disagreements in the argumentation game. We consider that, in order to set up an experiment, we always start with a associations $U \mapsto S$ that is both consistent, and homogeneous. Notice that this describe the ML data-set from the Irvine repository that we will be using in the experimental evaluation of our approach.

As experimenters, we will give a set of associations $U_1 \;{}_{o}\!\!\mapsto S_1$ to an agent A_1 and a set of associations $U_2 \;{}_{o}\!\!\mapsto S_2$ to an agent A_2 such that:

- $U \supseteq (U_1 \cup U_2)$, and
- $U_1 \;{}_{o}\!\!\mapsto S_1 \neq U_2 \;{}_{o}\!\!\mapsto S_2$.

Since the set of associations $U \mapsto S$ is both consistent and homogeneous, if $U_1 \;{}_{o}\!\!\mapsto S_1$ and ${}_{o}\!\!\mapsto S_2$ were both subsets of $U \mapsto S$, their union would also be consistent and homogeneous. This means that any example from U_1 or U_2 would be associated to exactly one sign s by A_1, and exactly one sign s' by A_2. The homogeneity of the two sets implies that A_1 and A_2 will learn the same classifications, therefore it also means that, under the condition that $S_1 = S_2$, the sign s and s' used to name e are the same sign. Therefore, if $U_1 \;{}_{o}\!\!\mapsto S_1$ and ${}_{o}\!\!\mapsto S_2$ were both subsets of $U \mapsto S$, the union of these two sets of associations would not (under the condition that $S_1 = S_2$) result in any disagreement between the agents in a naming game that takes place over a subset of $U_1 \cup U_2$. We call this setup the *Homogeneity* setup. We will now see how we can alter the Homogeneity setup such that the two sets of associations $U_1 \;{}_{o}\!\!\mapsto S_1$ and $U_2 \;{}_{o}\!\!\mapsto$ are transformed into two new sets $U_1 \;{}_{o'}\!\!\mapsto S_1'$ and $U_2 \;{}_{o'}\!\!\mapsto S_2'$ that will be distributed to the agents and will cause disagreements during the naming game over $U_1 \cup U_2$.

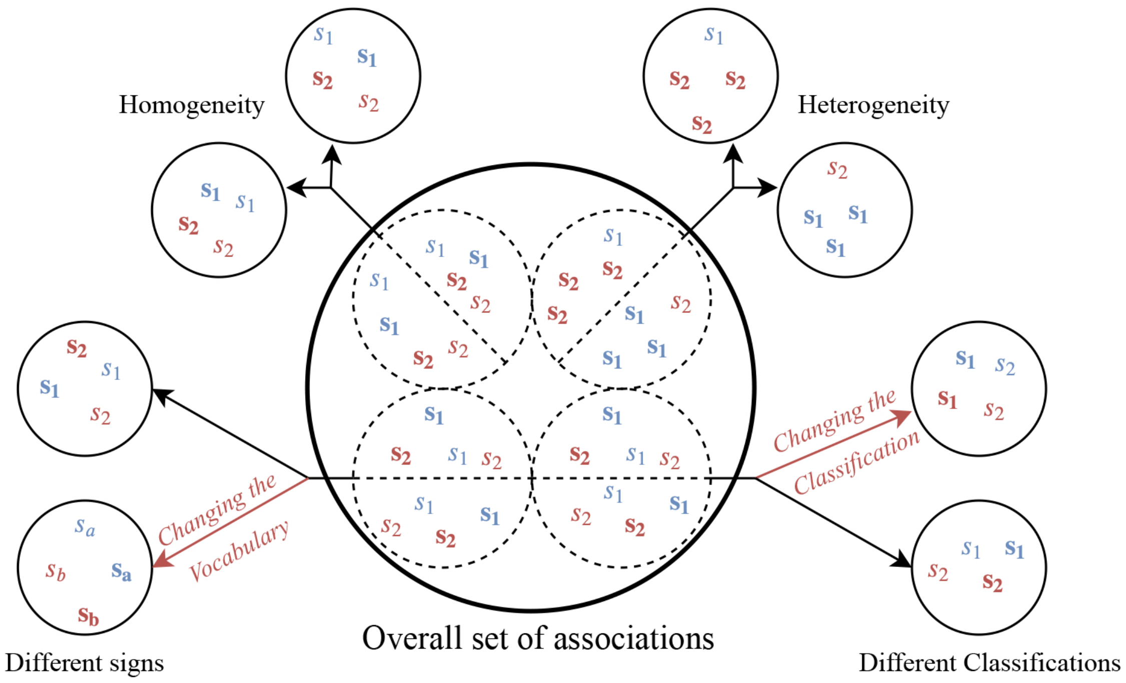


Figure 3.3: This figure illustrate how subsets of a same context can lead to different situations. Each example is represented by its sign (s_1 or s_2), and can have two features (color and boldness). In the overall set of associations, the blue examples are labeled s_1 while the red examples are labeled s_2. Actions that affect the labels of the examples are represented by red arrows.

3.4.1 Experimental Setups Explored

We are now going to explore the different experimental setups for a naming game between the agents A_1 and A_2. As an experimenter, we have the ability to alter all the aspects of the consistent and homogeneous set of associations $U_{o}{\mapsto}S$ before distributing its examples and signs among the agents, including:

- creating new signs,
- altering the homogeneity of subsets of $U \mapsto S$, and even
- choosing which signs are associated to which examples.

Figure 3.3 represents the four different setups that our model is covering, and that are presented in the rest of the section is this order: Homogeneity setup, Heterogeneity setup, Different signs setup, and Different classifications setup. The first one, the Homogeneity setup, has already been presented at the beginning of this Section. The three other setups are setups that will cause disagreements during the naming game.

Homogeneity Setup The situation is homogeneous when we distribute the consistent set of associations $U_{o}{\mapsto}S$ by giving to A_1 and A_2 different subsets of $U_{o}{\mapsto}S$ which share the same lexicon. While it is hard to guarantee that a set is homogeneous, concrete implementations of this setup can involve the two subsets receiving associations from each class of $U_{o}{\mapsto}S$ in similar proportions, which helps to obtain a high degree of homogeneity. The generalizations that will be learned over sets of examples of similar sizes are likely to be similar themselves, and we can expect to have $U_{A1}^{\ l}{\mapsto}S \approx U_{A2}^{\ l}{\mapsto}S$.

Heterogeneity Setup The setup is heterogeneous when we distribute the set of associations $U\ {}_o{\mapsto}S$ by giving to A_1 and A_2 different subsets of $U\ {}_o{\mapsto}S$, while ensuring that the degree of heterogeneity between the two sets increases. Concrete implementations of this setup would for instance see the classes of $U\ {}_o{\mapsto}S$ being distributed in different proportions among the sets given to the agents. In this setup, the generalizations are likely to be different from one agent to another, even for a same class, due to the agents learning with some degree of error concerning the accuracy of the concept descriptions and therefore likely to cause a situation where $U\ {}_{A1}^{\,l}{\mapsto}S \neq U\ {}_{A2}^{\,l}{\mapsto}S$ (some disagreements occur).

Different Signs Setup In the Different Signs setup, we do not alter the homogeneity of the sets of associations received by the agents, but instead we alter their lexicon. We start by distributing the associations of the consistent set of associations $U\ {}_o{\mapsto}S$ among two sets of associations, $U_1\ {}_o{\mapsto}S$ and $U_2\ {}_o{\mapsto}S$. However this time, before giving these sets of associations to the two agents directly —as it was the case in the Homogeneity setup–, the experimenter will replace the lexicon S of these sets by two new separate lexicons S_1 and S_2. In this situation, the classes are not modified: if the sign $s \in S$ has been replaced by the sign $s_k \in S_k$, then the equality $U({}_o{\mapsto}s) = U({}_o{\mapsto}s_k)$ holds. The fact that $U_1({}_o{\mapsto}s_1) \neq U_2({}_o{\mapsto}s_2)$ is due to $U_1 \neq U_2$, not to $S_1 \neq S_2$. In this situation, the set of all associations $(U_1\ {}_o{\mapsto}S_1) \cup (U_2\ {}_o{\mapsto}S_2)$ is not consistent, but each of its examples is associated with exactly one pair of signs: one sign from S_1 and one sign for S_2. Moreover, if an example from the set $(U_1\ {}_o{\mapsto}S_1) \cup (U_2\ {}_o{\mapsto}S_2)$ is associated with two signs $s_i \in S_k$ and $s_j \in S_{-k}$, then there will not be any another example e' from the same set associated to a pair of signs $s_i \in S_k$ and $s'_l \in S_{-k}$ such that $s_j \neq s_l$.

Different Classifications Setup In the situation of Different Classifications, we start again by separating the set of associations $U\ {}_o{\mapsto}S$ in two different subsets $U_1\ {}_o{\mapsto}S_1$ and $U_2\ {}_o{\mapsto}S_2$ and then we alter the associations each of these subsets individually. While this was also the case in the Different Signs setup, this time the modification is done in such a way that there is no relation of equivalence between the classes of $U_1\ {}_o{\mapsto}S_1$ and $U_2\ {}_o{\mapsto}S_2$. In this setup, the set $(U_1\ {}_o{\mapsto}S_1) \cup (U_2\ {}_o{\mapsto}S_2)$ would remain inconsistent for any modification of the lexicons S_1 and S_2. In concrete implementations of this setup, a set of associations $Q = U_k\ {}_o{\mapsto}S_k$ from an agent A_k is chosen, and two sets of associations $U_a \mapsto s$ and $U_b \mapsto s'$ are selected from Q. From there, we group the examples from $U_a(\mapsto s)$ and $U_b(\mapsto s')$ in one set of examples $U_c = U_a(\mapsto s) \cup U_b(\mapsto s')$. A sign $s'' \in \{s, s'\}$ is selected, and used to create a new set of associations $U_c \mapsto s''$. Then, we replace the two sets of associations $U_a(\mapsto s)$ and $U_b(\mapsto s')$ by the set of associations $U_c \mapsto s''$ in Q. This process creates two polylexematic classes in the overall set of associations $U_1\ {}_o{\mapsto}S_1 \cup U_2\ {}_o{\mapsto}S_2$, which will then cause disagreements. This change implies the disappearance of either s or s' from S_k, but the lexicons S_1 and S_2 still remain fairly similar.

Then, the resulting sets of associations are given to the agents as $U_1\ {}_o{\mapsto}S_1$ and $U_2\ {}_o{\mapsto}S_2$. In this setup, the fact that $(U_1\ {}_o{\mapsto}S_1) \cup (U_2\ {}_o{\mapsto}S_2)$ is inconsistent is enough to secure disagreements during the naming game, even if the lexicons S_1 and S_2 are similar.

3.5 Reaching Mutual Intelligibility

We have now presented the different types of scenarios that our learning agents are expected to encounter during our experiments. The details of our approach to reach mutual intelligibility in scenarios where they encounter disagreements is addressed in Chapter 4, and a model that uses our approach is proposed in Chapter 5. This model is later exemplified in Chapter 9 and experimentally evaluated in Chapter 10. However, we can already discuss generalities of our approach on how two agents can modify their concepts in order to reach mutual intelligibility. Our approach is the following: when confronted to a setup where they find disagreements, the agents aim is to change this setup to reach a Homogeneity setup, where no disagreement arise. As mentioned in introduction of this chapter, no disagreement means overall agreement in the naming game, and overall agreement means mutual intelligibility. We mentioned in the Section on Homogeneity Setups that it had to take steps toward creating a homogeneous set. In the rest of this chapter, however, we will still consider that a perfect homogeneity can be attained for the purpose of our explanations.

3.5.1 Reaching mutual intelligibility according to the situation

We will now address how the agents, with a limited set of actions at their disposal, can modify the three problematic setups into a Homogeneity setup. Intuitively, the agents can do three kinds of actions: create new signs, change which signs are associated to which examples in their individual sets of associations *only*, and communicate with each other exchanging relevant information.

Heterogeneity

The heterogeneity setup is caused by a lack of homogeneity. The setup of heterogeneity induces *overlaps* between the left-path associations of the two agents. Overlaps are another name for sets of examples that are generalized by concepts with different signs. These overlaps are due to undergeneralizing (with respect to the set of all examples) during learning using the individual agents examples. In this setup, the agents could just exchange all of their example-sign associations to recreate the original set of associations $U\ {}_{o}\!\mapsto\! S$, which is the union of their two local sets. Since $U\ {}_{o}\!\mapsto\! S$ is consistent, the agents could learn new concepts thatwould classify accurately since the context $U_1 \cup U_2$ is consistent and now shared by both agents. Indeed, here the set of associations $U_1\ {}^{l}_{1}\!\mapsto\! S$ and $U_2\ {}^{l}_{2}\!\mapsto\! S$ would be both equal to $U\ {}_{o}\!\mapsto\! S$, $U_1\ {}^{l}_{1}\!\mapsto\! S$ and $U_2\ {}^{l}_{2}\!\mapsto\! S$.

Of course, transferring all the examples from one agent to another is costly in terms of information (exchanging all examples). The agents have no control over the homogeneity degree of their sets of associations other than by exchanging examples, and if they want to limit the number of examples that they exchange they would need methods (such as AMAIL's argument exchange) to summarize the information of chosen subsets of their local sets of associations, expressed as arguments, and exchange them —additionally to exchanging (a much lower quantity of) example-sign associations.

Different signs

The Different Signs Setup is radically different the Homogeneity and Heterogeneity setups. If the agents were to exchange their example-signs associations in order to regroup their local sets of associations, they would obtain a set of associations with polylexematic classes that cannot be used for supervised learning. This time, the agents cannot simply use the AMAIL approach, since using AMAIL as a substitute to example-sign associations transfer would convey the same conflicting information between the agents as the example-sign associations transfer would.

A new approach is required. Thus, the agents can find which of their concepts are equivalent while having different signs, and change these signs for a same new sign. Since the agents receive different examples from the experimenter, the agents also have different classes, but they can still find equivalences between them through simple communication, as we will see later in Chapter 5. Communicating with each others, the agents can understand which concepts generalize the same examples from the context $U_1 \cup U_2$, and group pairs of concepts $(U_1 \cup U_2)\ {}^{l}_{A1} s_i$, $(U_1 \cup U_2)\ {}^{l}_{A2} s_j$ that are considered equivalent. We call this process a *mapping* of the classes of the two agents.

Once the concepts are mapped, the agents can change the signs of each pair of concepts so they both generate the same left-path associations. Once this is done, the agents are finally using the same lexicon. Upon aligning their right-path associations on their left-path associations, the union of the two agents' sets of associations will become a consistent set, and the agents will be in a Homogeneity setup and mutual intelligibility.

Different classifications

The Different Classifications setup is comparable to the Different Signs setup in the sense that the union of the two local set of associations of the agents cannot make a consistent set of associations in both setups. However, unlike the Different Signs setup, there is no possible one-to-one mapping that can draw equivalences between concepts of the agents and no substitution of a lexicon to another that can result in a Homogeneity setup.

According to Section 3.4, this situation is the result of a two-steps setup: first, the experimenter distributes the associations of a consistent set of associations $U\ {}_{o}{\mapsto}S$ among two subsets $U1\ {}_{o}{\mapsto}S$ and $U2\ {}_{o}{\mapsto}S$, creating a Homogeneity setup. Then, the experimenter groups different classes in each of these two sets in order to obtain two new sets $U1\ {}_{o'}{\mapsto}S'_1$ and $U2\ {}_{o'}{\mapsto}S_{2'}$ with different classes. In Different Classifications setups, the agents reach homogeneity by doing the opposite as the latter step of the setup. Instead of grouping sets of associations to create polylexematic classes, they will separate the polylexematic classes of the sets of associations $U1\ {}_{o}{\mapsto}S_1$ and $U2\ {}_{o}{\mapsto}S_2$ and achieve a Homogeneity setup. In order to separate the polylexematic classes of $U1\ {}_{o}{\mapsto}S_1$ and $U2\ {}_{o}{\mapsto}S_2$, the agents will create the Cardinal product $S_1 \times S_2$ of their lexicons. Then, the agents can again communicate to determine, for each pair of signs s, s' in $S_1 \times S_2$, if the polylexematic class $(U_1 \cup U_2)(\mapsto \{s, s'\})$ is empty. Once the non-empty polylexematic classes are identified, the agents are able to create a concept for each of them.

Next, the agents will split the classes of their sets of associations in order to reach a state of homogeneity. Their new classes should now match the classes of the sets of associations $U1\ {}_{o}{\mapsto}S_1$ and $U2\ {}_{o}{\mapsto}S_2$. The agents can then learn new concepts for these split classes, but they will not have yet achieved a Homogeneity setup: they still have different lexicons. However, the agents have reached a situation of the type we called Different Signs setup. From then on, the agents can achieve a homogeneous setup using the method described in Section 3.5.1. Once the agents have returned to a Homogeneity setup, they can learn new concepts from their classes and reach mutual intelligibility.

Setup Hybridisation

Of course, nothing limits the experimenter to the use of a pure type of setups. The setups can be hybridised in order to create more complex scenarios for the agents to reach mutual intelligibility. For instance, the experimenters can reduce the homogeneity of the agents' sets of associations before changing their lexicons, creating a hybrid scenario between a Different Sign setup and a Heterogeneity setup. Different classifications and Different Signs setup cannot coexist —one requires the existence of a one-to-one mapping between concepts while the other requires that their are none—, and therefore hybrid setups can only involve a heterogeneity setup and either a Different Classifications or a Different Signs setup. In order to present how the agents can resolve hybrid setups, we are going to illustrate how the agents can deal with the most complex scenario, which is the hybridisation of a heterogeneity setup with a Different Classifications setup. In this situation, the agents will adopt a strategy that consists into reaching mutual intelligibility over small parts of their overall context once at a time. They do so by moving a part of the hybrid setup to a Different Classifications setup, where they can therefore reach a contextual agreement by using the method described in Section 3.5.1.

First, the agents will realise the Cartesian product of their lexicons in order to identify the polylexematic classes, which are overlaps, and determine which new concepts need to be created. We give more details on this task in hybrid setups in Section 3.5.1. Next, since the agents cannot create satisfying concepts in heterogeneous setups, the agents will isolate two concepts that are causing

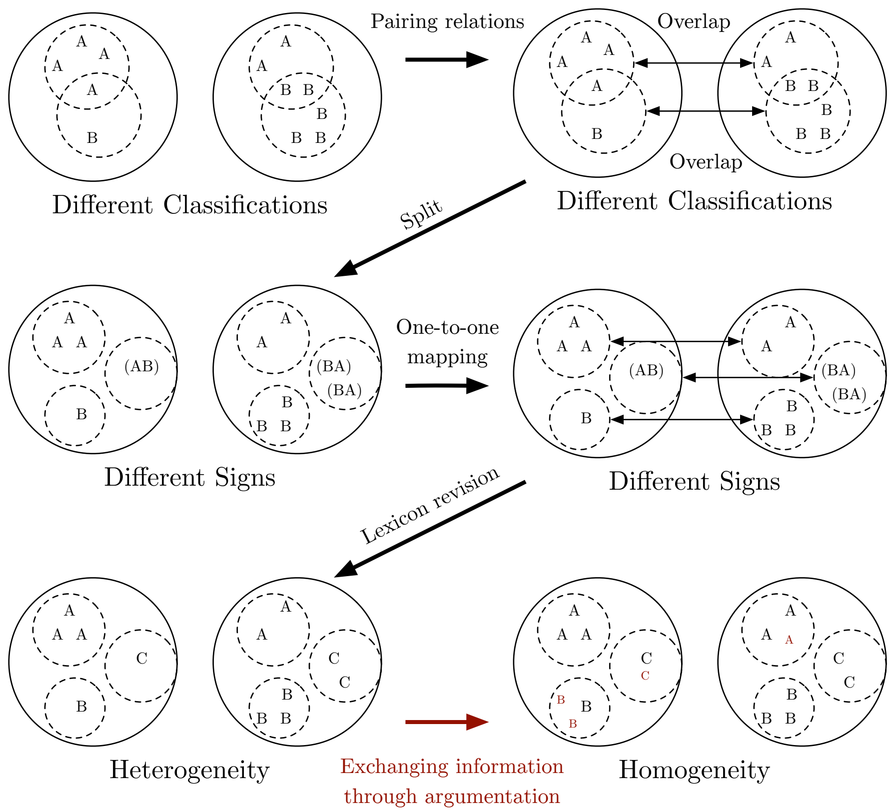


Figure 3.4: The strategy is moving from each situation to a simpler situation; the arrows name the method used.

an overlap and exchange enough information for the set of examples generalized by either concepts —we will call it U'— to be regarded as a homogeneous context. In the context U', the agents are in a different classification setup. The agents therefore create new concepts for each overlap (finding generalization using inductive learning), as explained in Section 3.5.1, and reach a Different Signs setup over U' before finally reaching a Homogeneous setup, again only in the context U'. The agents have now reached (partial) mutual intelligibility over U'. This whole process is illustrated in Figure 3.4. I need to change the figure by permuting some steps, it was badly explained before.

The Issue of Heterogeneity

In order to reach a Homogeneous setup, the agents need to know which strategy to use and therefore to understand in which setup or hybridisation of setups they are. For the moment, let's consider that there are no hybrid setups. The agents can now easily spot the Different Signs setup. Indeed, since each concept of an agent founds an equivalent concept in the other agent, the agents will notice that there are as many polylexematic classes as there are concepts. Heterogeneous setups are also easy to spot, as the agents will find polylexematic classes while using a same lexicon. Finally, if the agents are in neither of these two situations and still find disagreements, they know that they are in a Different Classification situation. Introducing the hybrid setups greatly complicates the task of identifying which type of situation is the case, mostly because of heterogeneity setups. Indeed, Different Classifications and Different Signs setup cannot coexist, therefore hybrid setups always involve heterogeneity.

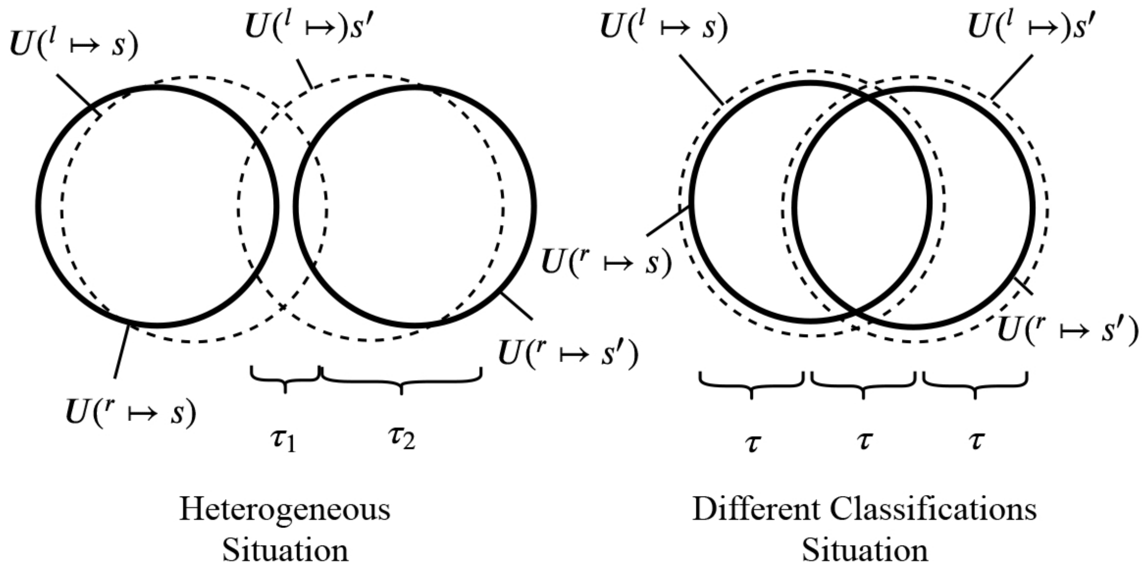


Figure 3.5: Graphic representation of the differences between the heterogeneous situation and the different classes situation. In each situation, two sets of right-path associations are resulting in two incoherent sets of left-path associations.

To illustrate the complexity that two agents faces when they need to identify situations while knowing that hybrid setups exist, let's take two concepts C_1 and C_2, the former created by an agent A_1 and the latter created by an agent A_2. While playing the naming game, the agents notice that there is a set of examples $U_{1,\overline{2}}$ that are associated to the sign $s = s(C_1)$ by A_1 and to another sign s_a, a set of examples $U_{1,2}$ that are associated to the sign s by A_1 and to the sign $s' = s(C_2)$ by A_2, and a third set of examples $U_{\overline{1},2}$ that are associated to the sign s_a, and to another sign s_b by A_1. The first thing that the agents can observe, is that they are not using the same lexicon. Therefore, they are not in a pure Different Signs setup or in a pure Heterogeneity setup. They might be, however, in a Different Classification setup. The sets $U_{1,\overline{2}}$, $U_{1,2}$ and $U_{\overline{1},2}$ are overlaps. They may have been caused by either:

- an over-generalization of C_1 and C_2 to the sets of examples $U_1(_o{\mapsto}s)$ and $U_2(_o{\mapsto}s)$ in a heterogeneous setup, that was then hybridised with a Different Signs setup by changing the sign s for s' in $U_2(_o{\mapsto}s)$, or
- a different classification setup in which there were three sets of associations $U_x(\mapsto x)$, $U_y(\mapsto y)$ and $U_z(\mapsto z)$ that have been regrouped; $U_x(\mapsto x)$ and $U_y(\mapsto y)$ regrouped into $U_1(_o{\mapsto}s)$, and $U_y(\mapsto y)$ and $U_z(\mapsto z)$ being regrouped into $U_2(_o{\mapsto}s')$.

In the former case, the agents should change the sign of one of the two concepts for the sign of the other, and then exchange information about their contexts in order to be able to learn new concepts to replace C_1 and C_2 once the agents are in a Homogeneity setup. In the latter case however, the agents cannot reach a Homogeneity setup by exchanging information as they are in a Different Classification setup. As shown in Figure 3.5, in order to identify a situation type, the agents need to decide of an fixed threshold value for the sizes of $U_{1,\overline{2}}$, $U_{1,2}$ and $U_{\overline{1},2}$, in such a way that above this value these three overlaps are not seen as the results of under-generalized concept learning but as three separated concepts that should be individually created.

3.6 Conclusion

The first requirement for an agent in our approach is to have a concept representation that makes a clear distinction between the left and right path associations. In both cases, the generalizations and examples should be represented separately from the signs they are associated with. Therefore, our first step in the definition of our approach will be the presentation of a semiotic model of concept representation, which is done in Section 4.2.1. The organization of these semiotic model of concepts into collection of concepts that partition a context is the next logical step, that is detailed in Section 4.2.2.

Once the concept collections of our agents has been established, arises the issue of finding a mapping between the two partitions they induce in the two individual agents. Our method to link two different concepts with each others is presented in Section 4.4.1. Mapping the relations between concepts is however not enough to reach mutual intelligibility. The agents need a clear protocol that specifies, for each type of mapping, in which situation the agents are. These situations are detailed in Section 4.5. As we mentioned in the previous subsection, the two agents cannot differentiate a situation of Heterogeneity from a situation of Different Classifications. For clarity's sake, we will first present a simplified model of argumentation that does not consider Heterogeneous situations, in Chapter 5.

Taking into account the possibility of heterogeneous situations is the same as taking into account a degree of type-one and type-two errors in the creation of the generalizations through inductive learning. Acknowledging the existence of errors during the inductive learning in our model, and how to address it, is explained in the Chapter 6. After that, the model will be ready for testing and evaluation on all scenarios involving both artificial and real data sets.

Chapter 4

An Approach to Mutual Intelligibility

4.1 A Semiotic Approach of Agents and Communication

4.1.1 An Overview of Feature Terms

Our protocol of argumentation is based on the capacity of agents to associate a semiotic elements from any type with a given set of examples – the adjunct set. While this capacity can be granted by various approaches in machine learning – as long as they can be interpreted as generalizing over and classifying examples, we will mostly focus on the use of feature terms to associate the semiotic elements with their adjunct sets.

Feature terms, also called feature structures or ψ-terms, are a generalization of first-order terms that have been introduced in theoretical computer science in order to formalize object-oriented capabilities of declarative languages (Aıt-Kaci, 2007) (Carpenter, 2005). Feature terms correspond to a different subset of firs-order logic than description logic, but have the same expressive power (Ait-Kaci and Podelski, 1993).

The example of feature terms presented bellow is taken from the journal paper *Similarity Measures over Refinement Graphs* (Ontañón and Plaza, 2012). Consider the apparently simple Trains data set shown in Figure 4.1, introduced by Michalski (Larson and Michalski, 1977): the original task is to learn the rule that discriminates east-bound from west-bound trains. If we were to represent such data set using a feature vector, we would need to define features for each one of the cars of a train (size, shape, load, and number of wheels), and determine beforehand a maximum number of cars per train (since feature vector representations have a fixed number of features).

Notice, however, that not all the trains have the same number of cars, and that, in principle, a train may have an unbounded number of cars. Thus, it is difficult to represent this data using a feature vector without losing information. Using a relational representation, we can just represent each car as a term, and define that a train is a set of cars, without restricting the number of cars of the train or the load each train is carrying.

On the other hand, Figure 4.2 represents the first west-bound train from Figure 4.1 using the feature term notation. We can see that the term is composed of 14 variables. The term contains two set- valued features (indicated by a curly bracket): in the feature cars of variable $X1$, and in the feature *lcont* of variable $X3$. Finally, we can also see that there are several variable equalities in this term. Since the value of the feature *infront* of variable $X2$ is the already defined variable

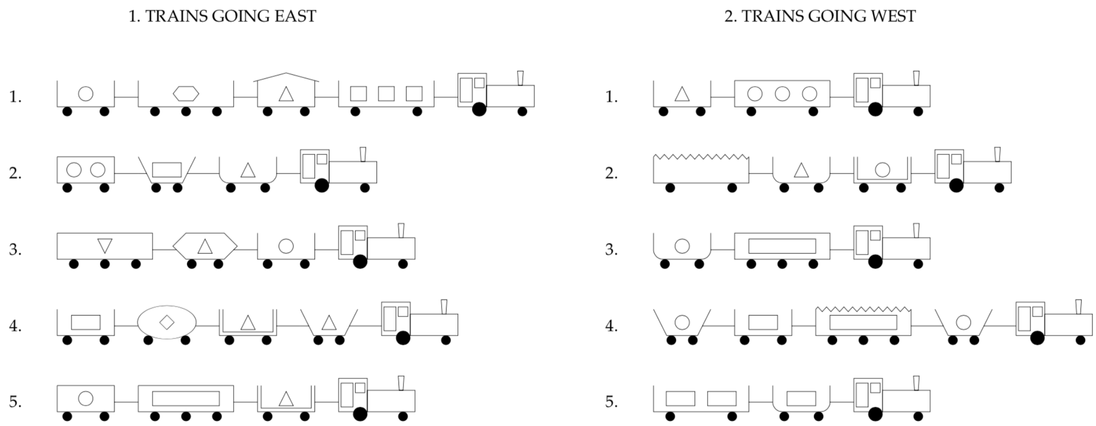


Figure 4.1: Graphic representation of a sample from the Trains data set introduced by Michalski. Each instance (train) has a different set of wagons, that display each a different set of properties.

$$\psi ::= X_1 : train \left[cars \doteq \left\{ \begin{array}{l} X_2 : car \begin{bmatrix} nwheels \doteq X_5 : 2 \\ ln \doteq X_6 : long \\ shape \doteq X_7 : engine \\ infront \doteq X_3 \end{bmatrix} \\ \\ X_3 : car \begin{bmatrix} nwheels \doteq X_5 \\ ln \doteq X_6 \\ shape \doteq X_8 : closedrect \\ lcont \doteq \left\{ \begin{array}{l} X_9 : circle \\ X_{10} : circle \\ X_{11} : circle \end{array} \right. \\ infront \doteq X_4 \end{bmatrix} \\ \\ X_4 : car \begin{bmatrix} nwheels \doteq X_5 \\ ln \doteq X_{12} : short \\ shape \doteq X_{13} : openrect \\ lcont \doteq X_{14} : triangle \end{bmatrix} \end{array} \right. \right]$$

Figure 4.2: First west-bound train from Figure 4.1 represented in feature-term notation.

$X3$, we note $X2.infront \doteq X3$, and also $X3.infront \doteq X4$. Additionally, the number of wheels in all the cars is the same, and the length of the first two cars is also the same.

The basic operation between feature terms is *subsumption*: we will use $\psi_1 \sqsubseteq \psi_2$ to express that a term ψ_1 subsumes another term ψ_2 – that is to say ψ_1 is more general (or equal) than ψ_2. Another interpretation of subsumption is that of an "informational content" order: $\psi_1 \sqsubseteq \psi_2$ means that all the information in ψ_1 (all that is true for ψ_1) is also contained in ψ_2 (is also true for ψ_2)[1].

4.2 Concepts and Contrast Sets

4.2.1 Semiotic Elements

The semiotic element are the components of concepts and containers. The different elements that an agent has perceived in its environment are called *examples*. For example, a specific bird is an example of *birds* or *animals*. Birds and animals are two *domains* in which we find birds. Domain is used here as a short for application domain. Each example is identified by an index i and noted e_i. Unless specified otherwise, agents are representing examples using a feature-term. A subset of a domain is a *context*, the examples from the domain that a given agent has knowledge of, which we define in Definition 7.

Definition 7 (Context)**.** *A context $U = \{e_1 \dots e_n\}$ is a set of examples that covers a large part of a domain.*

A context needs to have enough examples to be partitioned into several non-empty sets.

An agent can classify the examples of a context into sets of examples called *extensional definitions.* An extensional definition is associated to a specific category of the domain: for instance, the set of all birds of pray in a zoo can be an extensional definition for *bird of prey* in the context of the zoo's aviary, for the domain of birds. Definition 8 formalizes this.

Definition 8 (Extensional Definition)**.** *An extensional definition is a non-empty subset of a context: $E_i \subset U$ and $E_i \neq \emptyset$. Extensional definitions are semiotic elements.*

Some examples share similar features, and therefore can be generalized. A set of generalizations that generalize all the examples from an extensional definition is called an *intensional definition* (see Def.9). When the examples are represented as feature terms, the generalizations are also represented as feature terms that subsume sets of examples.

Definition 9 (Intensional Definition)**.** *Let $X \subset U$ be a subset of examples of a context U, an intensional definition of X is a set of generalizations $I_i = \{g_1, \dots, g_n\}$ such that $\forall e \in X, \exists g \in I_i$ such that $g \sqsubseteq e$, $\forall g \in I_i, \exists e \in X$ such that $g \sqsubseteq e$, and $\forall e' \in U - X, \nexists g \in I_i$ such that $g \sqsubseteq e'$. If an example e is subsumed by a generalization from I_i, we note $I_i \sqsubseteq e$. If E is a set of examples such that, for each $e \in E$, $I \sqsubseteq e$, we note $I \sqsubseteq E$. Finally, we define as $X_i = \{e \in U | I \sqsubseteq e\}$ the set of examples from U that are subsumed by I.*

When two agents are communicating, they are using *signs* (see Def.10). Signs are used to represent the labels of classes of examples. The knowledge representations that the agents use when they exchange examples or generalizations, feature terms for instance, are considered as represented by a system of signs – even if it is not explicitly detailed. Notice that there is no constraint on the sign, therefore the choice of a sign for a concept is arbitrary. The arbitrariness of the sign means that all of those signs are *symbols* from a semiotic point of view.

[1] In our work, we use the definition of subsumption introduced in (Arcos, 1997) which has a slightly different definition than the traditional *θ-subsumption*. Specifically, the difference is that we introduce the constraint that all the elements in a set have to be different.

Definition 10 (Sign). *A sign s_i is a string of characters. Signs are semiotic elements.*

The intensional, extensional definitions and the sign are the three primary semiotic elements. The relation between one sign, one intensional definition and one extensional definition is a *concept*. The concepts bind the ability to partition a context (extensional and intensional definitions) to the ability to express this partition through communication (intensional definitions and signs). Concepts are also the last type of semiotic elements. The notion of concept is defined in Definition 11.

Definition 11 (Concept). *A concept $C_i = (s_i, I_i, E_i)$ is a triadic relation between a sign, an intensional definition and an extensional definition. Given a context U and $X_i = \{e \in U | I_i \sqsubseteq e\}$, the relation should verify that $E_i = X_i$. We note $I_i \sqsubseteq E_i$ the fact that $\forall e \in E_i$, $I_i \sqsubseteq e$. If the concept C_i belongs to an Agent A_k, we note it C_i^k. Given an example e, if $I_i \sqsubseteq e$, we note $C_i \sqsubseteq e$. Concepts are semiotic elements.*

When the context allows no confusion, we may simply write s_i, I_i, E_i with the sub-index i indicating the concept C_i to which they belong. Otherwise, we will use the notation presented in Definition 12 to refer to the specific constituents of a concept C_i.

Definition 12 (Concept Constituents). *For any concept $C_i = (s_i, I_i, E_i)$*

1. $s(C_i) = s_i$
2. $I(C_i) = I_i$
3. $E(C_i) = E_i$

A same concept can be instantiated multiple times. For this reason, each concept C_i has an identifier id_i. Two instances of concepts that share their identifier are supposed to be instances of a same concept. The identifier of a concept C can also be noted $id(C_i)$. One agent is not suppose to have more than one instance of a concept, and therefore cannot have two concepts sharing the same identifier.

4.2.2 Containers And Contrast Sets

The agents are classifying the examples of their context. Each concept of an agent corresponds to a class, and the entire classification corresponds to a container. A container is the relation between a set of concepts and the context that this set of concepts is aiming to classify on, relation defined in Definition 13. There are two types of container: *hypotheses* (see Def.15) and *contrast sets* (see Def.14). Hypotheses are not partitioning their context, while Contrast sets do.

Definition 13 (Container). *A container $Q = (U_Q, S_Q)$ is a pair composed of a context U_Q and a set of concepts $S_Q = \{C_1, \ldots, C_n\}$. The notation $C_i \in Q$ means that the concept C_i belongs to the set of concepts S_Q, implying that $\forall C_i \in S_Q, E(C_i) \subset U_Q$.*

Contrast sets are a type of container where the extensional definitions of the concepts are a partition of the context. Contrast sets are defined as follow:

Definition 14 (Contrast Set). *A contrast set $K = (U_K, \{C_1, \ldots, C_n\})$ is a container where the set of the extensional definitions $\{E(C_1), ..., E(C_n)\}$ is a partition of the context U_K. This is noted as $\Pi(U_K) = E(C_1), \ldots, E(C_n)$. Moreover, the signs of the concepts must be different: $\forall C_i, C_j \in K$, $i \neq j \Rightarrow s(C_i) \neq s(C_j)$.*

Recall that a partition of a set S is a set of nonempty subsets of S such that every element $e \in S$ is in exactly one of these subsets.

Hypotheses, as contrast sets, do not have more than one concept with the same size. However, unlike contrast sets, the extensional definitions of their concepts can have any relations as long as they remain compliant with the definition of containers.

Definition 15 (Hypothesis)**.** *A hypothesis $H = (U_H, \{C_1, \ldots, C_n\})$ is a container where the signs of the concepts are different: $\forall C_i, C_j \in H$, $i \neq j \Rightarrow s(C_i) \neq s(C_j)$.*

The agents are using contrast sets to know which sign to use in order to refer to one example from a domain. Hypotheses are used by agents to build a copy of other agents contrast sets with their own context in order to try to understand the point of view of other agents.

Concepts can be noted alternatively by using their signs or their identifiers. A concept from a contrast set or a hypothesis can be noted by using its sign and container, as two concepts from a same contrast set or hypothesis cannot share the same sign. Identifiers can be used to find a concept from any container of an agent, as an agent cannot have two concepts with the same identifier. This notation is useful to refer to a concept as a container's constituent, and is presented in Definition 16 below:

Definition 16 (Container Constituent)**.** *Let A_1 be an agent with n containers $Q_1 = (S_1, U_1)$, ..., $Q_n = (S_n, U_n)$ and C_i a concept such that $C_i \in S_i$. If $s = s(C_i)$ and $id = id(C_i)$, therefore:*

$$C_i = C(s, Q_i) = C(id, A_1).$$

4.3 Agents

Knowledge Our approach focuses on pairs of agents that communicate together. Each agent is able to represent its knowledge with semiotic elements arranged in concepts, which are themselves arranged in containers. This allows the agents to have a partition of their contexts (the extensional definitions), a set of symbols to communicate over the examples of this partition (the signs) and generalizations to either incorporate new examples to their partition or access the reflexive function of language and address the meaning of their concepts in their communications with the other agent.

The two agents are numbered, called A_1 and A_2. Any agent can be referred as A_k, and in this case the other agent is referred as A_{-k}. An agent A_k has knowledge over three containers: two contrast sets and one hypothesis. Among these two contrast sets, we distinguish the initial contrast set K_k^0 of an agent from the contrast set in use to communicate with the other agent K_k, called the *current* contrast set. When an agent A_k has a concept C in its current contrast set, we say that A_k *knows* C.

The hypothesis H_k of A_k shares its context with K. H_k is used to store the information that A_k has on the other agent's concepts. As mentioned in Section 4.2.2, hypotheses are used by agents to build a copy of other agents contrast sets with their own context in order to try to understand the point of view of other agents.

In general, any container Q attached to an agent A_k will be noted Q_k while its context will be noted U_k (see Section 4.2.1). A semiotic element x attached to A_k will be noted x^k.

Messages An agent can exchange information with the other agent using messages. A message $M(x_1, \ldots, x_n)$ contains n semiotic elements. The letter M is a place-holder for a *performative*, that helps the agent that receives the message to understand what it is supposed to do with it.

For instance, an agent receiving a message *Examples(*e_1, e_2, e_3*)* knows that the other agent wants it to add the examples e_1, e_2 and e_3 to its contrast set's context.

By convention, when an agent A_k wants to refer to two concepts in a message, one concept C_i from its contrast set K_k and another concept C_j from the other agent's contrast set A_{-k}, A_k starts its message with two signs: $s(C_i)$ first and $s(C_j)$ in second.

Functions The central function of an agent is to name an example when one is presented to it. In order to name an example e, an agent A_k searches in its *current* (not initial) contrast set $K_k = \{U_k, S_k\}$ for the set of signs $N(A_k, e) = \{s(C)|C \in S_k \wedge I(C) \sqsubseteq e\}$ from the concepts of S_k that are subsuming e. A_k names the example e with $N_k(e)$.

An agent has also access to various other functions that allow it to organize its contrast set and argue about it with another agent. These functions are presented later in detail in 5.1.

Synchronization In order to have a synchronized communication, two agents are using a token when they communicate together. There is only one token in a communication between two agents. When an agent receives the token, the agent can act, using one or more of its functions. Once it is done with its actions, the agent passes the token to the other agent and waits until it receives the token back. The time lapse between the moment one agent receives the token and gives it back is called a *turn*. The time lapse between two consecutive moments when one agent gets the token is called a *round*.

States An agent's state determines which actions the agent will use during its turn. The next action that an agent takes during its turn is to pass the token to the other agent, regardless of the state, at the exception of the state where the agent stops the conversation. The second to last action that an agent takes during its turn is to select the state of its next turn. States are described later in Section 5.3.3.

4.4 Relations between Concepts

4.4.1 Adjunct Sets and Relations

Adjunct Sets

We saw in Section 4.3 how the agents can name examples. For a given concept C and a given example e we can tell if an agent A_k, that has C in the set of concepts of its current contrast set, has $s(C)$ in the set of signs $N_k(e)$ used to name e. In order to do so, we just need to verify that $I(C) \sqsubseteq e$: if this is the case, then $e \in N_k(e)$. Otherwise, $e \notin N_k(e)$. If an agent A_k returns $N_k(e)$ with a sign $s \in N_k(e)$ while naming an example e, we say that A_k named e with s.

We want to represent, for a given context U and a given concept C, which examples from U would be named $s(C)$ by an agent that has C in its current contrast set. This brings us the notion of adjunct set defined below:

Definition 17 (Adjunct Set of a Concept)**.** *The adjunct set of a concept C in the context U is $Adj(C, U) = \{e \in U|C \sqsubseteq e\}$.*

As we can see in Definition 17, the adjunct set of a concept is independent of the sign or extensional definition of this concept. Indeed, only the intensional definition is needed to compute the adjunct set. Therefore, we can define the adjunct set of any set of generalizations as:

Definition 18 (Adjunct Set of an Intensional Definition). *The adjunct set of an intensional definition I in the context U is $Adj(I, U) = \{e \in U | I \sqsubseteq e\}$.*

Both adjunct sets and extensional definitions are sets of examples and subsets of contexts. However, they are different because extensional definitions are conceived as semiotic elements, while adjunct sets are an auxiliary notion conceived to compare concepts. An important property of adjunct set is that they are possible to link them to extensional definitions through the use of Property 4.

Property 4 (Extensional Definitions and Adjunct Set). *Let C be a concept in the context U, such that $C = (s, I, J)$. If $C = (s, I, E)$ is a concept in the context U, therefore $E = Adj(C, U)$.*

Proof. Let C be a concept in the context U, such that $C = (s, I, J)$. According to Definition 11, $E = \{e \in U | I \sqsubseteq e\}$. According to Definition 17, $Adj(C, U) = \{e \in U | I \sqsubseteq e\}$. Therefore, $E = Adj(C, U)$. □

Pairing Partial Sets

The notion of adjunct set allows us to compare concepts. With the adjunct sets of any pair of concepts C_i and C_j, it is possible to say for a given context U which examples from U would be named $s(C_i)$ and not $s(C_j)$, which examples from U would be named $s(C_j)$ and not $s(C_i)$, and which examples from U would be named both $s(C_i)$ and $s(C_j)$ by an agent that knows both concepts. In order to do so, we introduce the notion of pairing partial set. the three pairing partial sets represent the set relation between two adjunct sets from a same context. The notion of partial set is defined below:

Definition 19 (Pairing Partial Sets). *We define three sets that characterize any pair of concepts C_i and C_j: $U_{i,\bar{j}}$, $U_{i,j}$ and $U_{\bar{i},j}$. Theses three sets, called the pairing pairing partial sets, are defined as follows:*

1. $U_{i,\bar{j}} = Adj(C_i, U) - Adj(C_j, U)$
2. $U_{i,j} = Adj(C_i, U) \cap Adj(C_j, U)$
3. $U_{\bar{i},j} = Adj(C_i, U) - Adj(C_j, U)$

Let A_k be an agent that knows both C_i and C_j: while A_k would name $s(C_i)$ but not $s(C_j)$ the examples from $U_{i,\bar{j}}$, $s(C_j)$ but not $s(C_i)$ the examples from $U_{i,\bar{j}}$ and both $s(C_i)$ and C_j the examples from $U_{i,j}$.

Pairing Relations

In order to represent the information that is given by one pairing partial set, we introduce the notion of evaluation function that is defined in Definition 20 below.

Definition 20 (Evaluation Function). *Let ? be the token for an unknown value, I the index set $\{-1, 0, 1\}$, $\mathbb{X}$ the set of all possible concepts, $\mathbb{U}$ the set of all possible contexts and $\mathbb{F}$ the set of functions $\mathbb{U} \to \mathbb{N} \cup \{?\}$. The evaluation function $ev : I \times \mathbb{X} \times \mathbb{X} \times \mathbb{U} \times \mathbb{F} \to \mathbb{N} \cup \{?\}$ is the function that for each pair of concepts C_i, C_j, for a given index $x \in I$, a given context U and a given function $f \in \mathbb{F}$ yields:*

$$ev(x, C_i, C_j, U, f) = \begin{cases} f(U_{C_i,\overline{C_j}}), & \text{if } x = -1. \\ f(U_{C_i,C_j}), & \text{if } x = 0. \\ f(U_{\overline{C_i},C_j}), & \text{if } x = 1. \end{cases} \tag{4.1}$$

In order to aggregate the information that is given by all the partial sets, we now introduce the notion of r-triplet. A r-triplet is a mathematical representation of the three partial sets of a given pair of concepts for a given context. This mathematical representation is a triplet of bits, each bit representing whether or not a given pairing partial set is empty.

Definition 21 (R-Triplet Function)**.** *Let $f(U_x)$ be the function that, for every pairing partial set U_x yields:*

$$f(U_x) = \begin{cases} 1, & \text{if } |U_x| \geq 1. \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

Let ev be the function defined in Definition 20. The r-triplet function $r : \mathbb{X} \times \mathbb{X} \times \mathbb{U} \to \{0,1\}^3$, with $\mathbb{X}$ the domain of semiotic elements and $\mathbb{U}$ the domain of all possible contexts, is a function that for each pair of concepts C_i,C_j and for a given context U yields a triplet $r(C_i, C_j, U) = (b_{-1}, b_0, b_1)$, called r-triplet, such that for $x \in \{-1, 0, 1\}$:

$$b_x = ev(x, C_i, C_j, U, f).$$

There are therefore $2^3 = 8$ possible r-triplets between two concepts C_i and C_j in the context U. The r-triplet gives the *pairing relation* of C_i and C_j in U accordingly to Definition 22:

Definition 22 (Pairing Relations)**.** *The pairing relations between a pair concepts C_i and C_j in a context U, represented with the operator $C_i r_U C_j$ where $r \in \{\equiv, \oslash, \odot, \otimes, \dagger, \ominus\}$, are the following:*

- *if $r(C_i, C_j, U) = (0,1,0)$ they are in an* equivalence *relation, noted $C_i \equiv_U C_j$*
- *if $r(C_i, C_j, U) = (1,0,1)$ they are in is a* disjunction *relation, noted $C_i \oslash_U C_j$*
- *if $r(C_i, C_j, U) = (1,1,1)$ they are in is an* overlap *relation, noted $C_i \otimes_U C_j$*
- *if $r(C_i, C_j, U) = (1,1,0)$ or $r(C_i, C_j, U) = (0,1,1)$ they are in an inclusion relation, noted $C_i \odot_U C_j$*
- *if $r(C_i, C_j, U) = (1,0,0)$ or $r(C_i, C_j, U) = (0,0,1)$ they are in a one-sided relation, noted $C_i \dagger_U C_j$.*

Whenever $r(C_i, C_j, U) = (0,0,0)$, there is no pairing relation between the semiotic elements, which is noted $C_i \ominus_U C_j$.

According to Definition 22, there is no pairing relation when the adjunct sets of both C_i and C_j are empty. The pairing relation is one-sided when only one of the adjunct set is empty.

Each r-triplet has a corresponding *reverse*. Let the reverse of a r-triplet be defined as follows:

Definition 23 (R-Triplet Symmetry)**.** *Let the matrix J be the exchange matrix of size 3×3:*

$$J = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

The reverse of a r-triplet R is the matrix product RJ.

Let C_i and C_j be two concepts, and let $C_i r_U C_j$ be the pairing relation of C_i with C_j in U. Let $r(C_i, C_j, U) = R$ and $r(C_j, C_i, U) = R'$. Since Definition 19 shows that $R = R'J$ and $R' = RJ$, R and R' are the reverse of each other and therefore it is easy to see that each r-triplet gives the same pairing relation as its reverse r-triplet, $C_i r_U C_j = C_j r_U C_i$, therefore the operator r_U from Definition 22 is commutative.

4.4.2 R-Triplets and Associated Pairing Partial Sets

Let C_i and C_j be two concepts and U a context. The three pairing partial sets of C_i and C_j in U can alternatively noted $U(x, C_i, C_j)$ with $x \in \{-1, 0, 1\}$ such that:

- $U(-1, C_i, C_j) = U_{C_i,\overline{C_j}}$,
- $U(0, C_i, C_j) = U_{C_i,C_j}$ and
- $U(1, C_i, C_j) = U_{\overline{C_i},C_j}$.

The elements a, b and c of a r-triplet $r = r(C_i, C_j, U) = (a, b, c)$ can also be alternatively noted $r[x]$ with $x \in \{-1, 0, 1\}$ such that $r[-1] = a$, $r[0] = b$, and $r[1] = c$. According to Definition 21, the element $r[x]$ or the r-triplet r is equal to $ev(x, C_i, C_j, U, f)$. Moreover, according to Definition 20, we have:

$$ev(x, C_i, C_j, U, f) = f(U(x, C_i, C_j)).$$

Therefore, it is possible to express the value $r[x]$ of a r-triplet according to a pairing partial set $U(x, C_i, C_j)$. We say that $r[x]$ is the associated value of $U(x, C_i, C_j)$, and that $U(x, C_i, C_j)$ is the associated pairing partial set of $r[x]$.

4.4.3 Overall Pairing Relations

Let A_1 and A_2 be two agents, A_1 partitioning the context U_1 in the contrast set $K_1 = \{U_1, S_1\}$ and A_2 partitioning the context U_2 in the contrast set $K_2 = \{U_2, S_2\}$. If the agents want to compare two concepts $C_i \in S_1$ and $C_j \in S_2$, they can find the adjunct sets of both concepts, compute the corresponding pairing partial sets, use the function r to find the r-triplet and deduce the pairing relation. However, the first element in the chain that allows the deduction of the pairing relation is the adjunct set. Since adjunct sets are context-dependent, the agents need to decide in which context they want to compare C_i and C_j.

In an experimental set-up, there are three main contexts that are of interest. The two first context are U_1 and U_2, that are referred as the *local* contexts, and the last one is the *overall* context $U_1 \cup U_2$ presented in Definition 24. Since two pairing relations in different local contexts might be different, and since the overall context includes both U_1 and U_2, the agents can agree that the overall r-triplet $r(C_i, C_j, U_O)$ is obtained from more information than they currently dispose to compute their local r-triplets and therefore should supplant their local r-triplets. C_i and C_j in the overall context.

Definition 24 (Overall Context)**.** *The overall context $U_O = U_1 \cup U_2$ of two containers $Q_1 = (U_1, S_1)$ and $Q_2 = (U_2, S_2)$ is the union of their contexts U_1 and U_2.*

In order to find an adjunct set $Adj(C, U_O)$, the agents can exchange their respective adjunct sets $Adj(C, U_1)$ and $Adj(C, U_2)$ since $Adj(C, U_O) = Adj(C, U_1) \cup Adj(C, U_2)$. However, this solution requires for the agents to exchange the adjunct sets. Since the adjunct sets can contain many examples, this represents a heavy transfer and is not the privileged solution. Moreover, according to Conjecture 1, an overall pairing partial set is the union of its two local pairing partial sets. The agents can therefore directly find an overall pairing partial set by exchanging their local pairing partial set.

Conjecture 1 (Overall Pairing Partial Sets)**.** *Let A_1 and A_2 be two agents, A_1 partitioning the context U_1 in the contrast set $K_1 = (U_1, S_1)$ and A_2 partitioning the context U_2 in the contrast set $K_2 = (U_2, S_2)$. Let C_1 and C_2 two concepts such that $C_1 \in S_1$ and $C_2 \in S_2$. For any $x \in \{-1, 0, 1\}$, we have:*

$$U_O(x, C_1, C_2) = U_1(x, C_1, C_2) \cup U_2(x, C_1, C_2).$$

We have not finalized the formal proof of this conjecture, but will explain the main ideas behind it. The schema of the proof would be the following: we show that each pairing partial set $U_k(x, C_1, C_2)$ of a local context U_k is the subset of examples from U_k that verifies one of the following:

- $\Phi_{-1}(e) = C_1 \sqsubseteq e \wedge C_2 \not\sqsubseteq e$, if $x = -1$
- $\Phi_0(e) = C_1 \sqsubseteq e \wedge C_2 \sqsubseteq e$, if $x = 0$
- $\Phi_1(e) = C_1 \not\sqsubseteq e \wedge C_2 \sqsubseteq e$, if $x = 1$

We show that the union of the set of examples from local context U_1 that verify a predicate $\Phi_x(e)$ with the set of examples from the local context U_2 that verify the same predicate $\Phi_x(e)$ is also the set of examples from the union $U_1 \cup U_2$ that verify the predicate $\Phi_x(e)$. We recall that according to Definition 24, the union $U_1 \cup U_2$ is the overall context U_O. Therefore, we may prove that the overall pairing relation $U_O(x, C_1, C_2)$, which is the set of examples from the overall context U_O that verify the predicate Φ_x, is equal to the union of the two local pairing partial sets $U_1(x, C_1, C_2)$ and $U_2(x, C_1, C_2)$. □

The agents have another solution, that allows them to find the pairing relation between C_i and C_j with far less data exchanged. Computing the local r-triplet $r(C_i, C_j, U_1)$ and $r(C_i, C_j, U_2)$ only requires from the agents to transfer the intensional definitions I_i and I_j, since with these two intensional definitions any agent A_k can compute the adjunct set $Adj(C_i, C_j, U_k)$ (see Definition 17). By exchanging their local r-triplets, the agents can finally infer the *overall* pairing relation using Conjecture 2.

Conjecture 2 (Expression of Overall Pairing Relation)**.** *Let A_1 and A_2 be two agents, A_1 partitioning the context U_1 in the contrast set $K_1 = (U_1, S_1)$ and A_2 partitioning the context U_2 in the contrast set $K_2 = (U_2, S_2)$. Let $C_i \in S_1$ and $C_j \in S_2$ be two concepts, and*

- *let $r(C_1, C_2, U_1) = (b_{-1}, b_0, b_1)$ be the local triplet of A_1*
- *let $r(C_1, C_2, U_2) = (b'_{-1}, b'_0, b'_1)$ be the local triplet of A_2*
- *let $r(C_1, C_2, U_O) = (b''_{-1}, b''_0, b''_1)$ be the overall triplet*

then, for all $n \in \{-1, 0, 1\}$, the element of the overall triplet $b''_n = 0 \Leftrightarrow b_n = 0 \wedge b'_n = 0$, and otherwise $b''_n = 1$.

We have not finalized the formal proof of this conjecture, but will explain the main ideas behind it. The schema of the proof would be the following: We show that the overall pairing partial set $U_O(x, C_1, C_2)$ is empty if and only if both local pairing partial sets $U_1(x, C_1, C_2)$ and $U_2(x, C_1, C_2)$ are empty. We show that if the pairing partial set $U(x, C_1, C_2)$ is empty, then the element with the sub-index x from its r-triplet is 0 according to Definition 21. Therefore, we may prove that an element b''_x of the overall r-triplet is equal to zero if and only if the elements b_x and b'_x of the local r-triplet are also equal to zero. □

Conjecture 2 results in four common and remarkable cases of local pairing relations for which the overall relation can be inferred just by transferring the two local pairing relations an not the local r-triplets. These four cases are listed below in Conjecture 3:

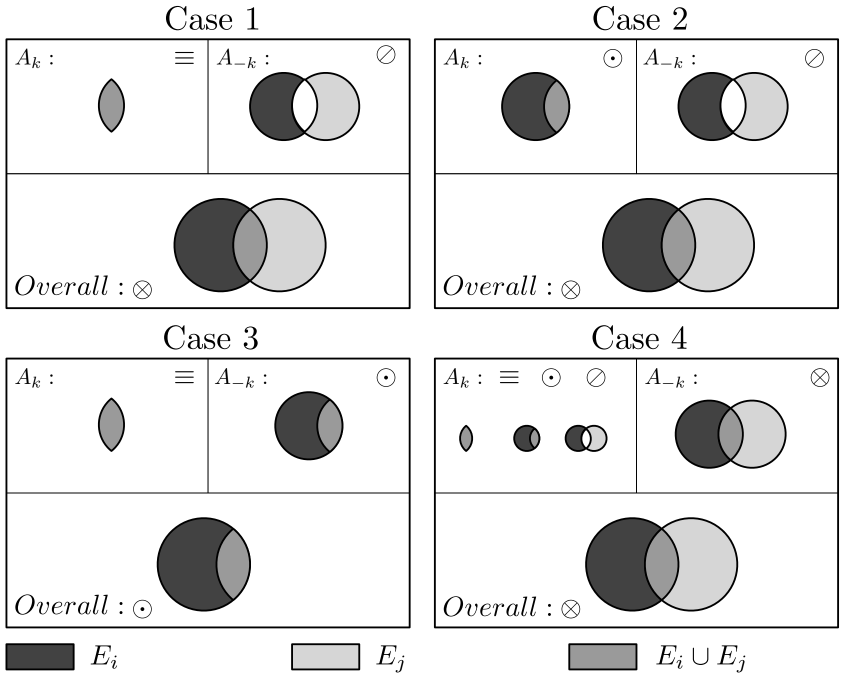


Figure 4.3: Venn diagrams illustrating of the four cases where it is possible to infer an overall relations' type from its two corresponding local relations' types.

Conjecture 3 (Overall Pairing Relation of Different Local Relations)**.** *Let A_1 and A_2 be two agents, A_1 partitioning the context U_1 in the contrast set $K_1 = (U_1, S_1)$ and A_2 partitioning the context U_2 in the contrast set $K_2 = (U_2, S_2)$. Let C_i and C_j be two concepts such that $C_i \in S_1$ and $C_j \in S_2$. Let $C_i r_{U1} C_j$ be the local pairing relation of the agent A_1 and $C_i r_{U2} C_j$ the local pairing relation of the agent A_2, if $C_i r_O C_j$ is the overall pairing relation of A_1 and A_2 between C_i and C_j, then the following holds:*

1. *if $C_i \equiv_{U1} C_j$ and $C_i \oslash_{U2} C_j$, then $C_i \otimes_O C_j$*
2. *if $C_i \odot_{U1} C_j$ and $C_i \oslash_{U2} C_j$, then $C_i \otimes_O C_j$*
3. *if $C_i \equiv_{U1} C_j$ and $C_i \odot_{U2} C_j$, then $C_i \odot_O C_j$*
4. *if $C_i \equiv_{U1} C_j$, $C_i \odot_{U2} C_j$ or $C_i \oslash_{U1} C_j$ and $C_i \otimes_{U2} C_j$, then $C_i \otimes_O C_j$*

We have not finalized the formal proof of this conjecture, but the proof would be a direct application of Conjecture 2 to the four listed scenarios. □

At last, since the r-triplets of the pairing relations of equivalence, disjunction and overlap are symmetrical (let R be a r-triplet that corresponds to a relation equivalence, disjunction or overlap. We can observe in Definition 22 that in these three cases, $R = RJ$), if two local pairing relations are both equivalences, disjunctions or overlaps then the overall pairing relation will be from the same type. For the remaining cases, the agents have no choice but to exchange their local r-triplets instead of their local pairing relations in order to infer the overall pairing relation.

4.5 Agreement and Disagreement

4.5.1 Mutual Intelligibility and Monotonicity

The argumentation on meaning revolves around mutual intelligibility. Mutual intelligibility is context dependent, and refers to a state of the multi-agent system where both agents are able to name every example from a given context with the same sign. Since the agents are not meant to go through an extensive naming game over an entire context outside of the experimental context of our thesis, mutual intelligibility refers also to a state where both agents agree that there is no example from a given context that they would – to their knowledge – name differently than the other agent.

The notion of mutual intelligibility is attached to two properties. First, both agents should partition the context of their mutual intelligibility into the same segregates, at least theoretically. Then, both agents should map these segregates to the same signs. Having these two properties, equal partition and equal sign-mapping for both agents, guarantees the mutual intelligibility between the agents over a given context. This notion of mutual intelligibility is formalized in Definition 25.

Definition 25 (Mutual Intelligibility). *Let A_1 and A_2 be two agents that have for respective contrast sets $K_1 = (U_1, S_1)$ and $K_2 = (U_2, S_2)$. A_1 and A_2 have reached mutual intelligibility under the limits of a context U whenever $\forall e \in U$, $!\exists C_i \in S_1$ and $!\exists C_j \in S_2$ such that $C_i \sqsubseteq e$ and $C_j \sqsubseteq e$ and $s(C_i) = s(C_j)$.*

Now that we stated multiple times that the ultimate goal of our agents is to reach mutual intelligibility and that we have formally defined this mutual intelligibility, we need to address the question of the mutual intelligibility's context. Since the mutual intelligibility is context dependant, the agents need know over which context they are trying to reach it in order to coordinate. We will assume that this context has examples from both agents, that are however not in both current contrast sets' context, which is the most complex possible scenario. In this scenario, each agent A_k could send to the other agent A_{-k} the set of the examples from U_k that are not in U_{-k}. However, this requires for each agent A_k to know which are these examples and thus have knowledge over U_{-k}.

Another solution that does not necessitate for A_k to have knowledge over U_{-k} is to know the pairing relations between the concepts that have examples in the context U over which the mutual intelligibility is wished. Conjecture 4 draws a direct correspondence between the definition of a mutual intelligibility over the context U and the pairing relations between concepts in U.

Conjecture 4 (Constraints on Relations for Mutual Intelligibility). *Let A_1 and A_2 be two agents with contrast sets $K_1 = (U_1, S_1)$ and $K_2 = (U_2, S_2)$. Let U be a context such that $U \subseteq U_O$ where all concepts of S_1 and S_2 subsume at least one example. We say that A_1 and A_2 have reached mutual intelligibility within the context U according to Definition 25 if for each concept $C_i \in S_k$ (for $k \in \{0, 1\}$) when the following holds:*

a. $!\exists C_j \in S_{-k}$ *such that* $C_i \equiv_U C_j$

b. ***and*** $\forall C_m \in S_{-k}$, *one of the two following holds:*

- *b1.* $C_i \equiv_U C_m$
- *b2.* $C_i \oslash_U C_m$

c. ***and*** $\forall C_m \in S_{-k}$:

- *c1. if* $C_i \equiv_U C_m$ *then* $s(C_i) = s(C_m)$
- *c2. if* $C_i \oslash_U C_m$ *then* $s(C_i) \neq s(C_m)$

We have not finalized the formal proof of this conjecture, but will explain the main ideas behind it. The schema of the proof would be the following: We prove that each concept C_i in the set of concept S_k having only one equivalent concept C_j in S_{-k} such that $C_i \equiv_U C_j$ is equivalent to $Adj(C_i, U)$ being equal to $Adj(C_j, U)$. We prove that $Adj(C_i, U)$ being equal to $Adj(C_j, U)$ is equivalent to each example e of U being covered by the concept C_i if and only if the concept C_j also covers e. Therefore, we prove that each concept C_i in the set of concepts S_k having only one equivalent concept C_j in S_{-k} such that $C_i \equiv_U C_j$ is equivalent to each example of U being subsumed by a unique pair of concepts (C_i, C_j) belonging to different contrast sets.

We prove that the adjunct set $Adj(C_i, U)$ being equal to the adjunct set $Adj(C_j, U)$ and C_j being unique is equivalent to each example e of U not being covered by any concept $C_m \in S_{-k}$ different from C_j. We prove that each example e of U not being covered by any concept C_m in S_{-k} different from C_j added to the fact that C_m cannot be empty is equivalent to C_i and C_m having the pairing relation $C_i \oslash_U C_m$. Therefore, we prove that each concept C_i in the set of concept S_k having only one equivalent concept C_j in S_{-k} such that $C_i \equiv_U C_j$ is equivalent to C_i having either a pairing relation $C_i \equiv_U C_m$ or a pairing relation $C_i \oslash_U C_m$, for all concepts C_m in S_{-k}.

We prove that having, for all concepts C_m in S_{-k}, either a pairing relation $C_i \equiv_U C_m$ and the signs $s(C_i)$ and $s(C_j)$ being equal, or a pairing relation $C_i \oslash_U C_m$ such that the signs $s(C_i)$ and $s(C_m)$ are different, is equivalent to each example of U being subsumed by each concept in a unique pair of concepts (C_i, C_j) belonging to different contrast sets with the sign $s(C_i) = s(C_j)$.

Therefore we prove that each example of U being subsumed by a unique pair of concepts C_i, C_j belonging to different contrast sets and $s_i \neq s_j$ is equivalent to each concept C_i in the set of concepts S_k having only one equivalent concept C_j in S_{-k} such that $C_i \equiv_U C_j$ and having for all concepts C_m in S_{-k}, either a pairing relation $C_i \equiv_U C_m$ and the sign $s(C_i) = s(C_j)$, or a pairing relation $C_i \oslash_U C_m$ and the signs $s(C_i) \neq s(C_m)$. $\square$

With Conjecture 4 the agents can now infer a mutual intelligibility over the context U from the overall pairing relations of the concepts involved in the context U. The agents can figure the overall pairing relations between concepts by exchanging r-triplets as explained in Section 4.4.3. Therefore, the agents do not need to exchange examples in order to know if they have reached a mutual intelligibility over a context – even if examples of this context are not shared by both agents.

While we repeatedly asserted that the aim of the argumentation between our two agents is to reach mutual intelligibility, the mutual intelligibility itself cannot be the only goal. We can intuitively think of two unsatisfying solutions that always guarantee a mutual intelligibility; the first one is having both agents' current contrast sets $K_1 = (U_1, S_1)$ and $K_2 = (U_2, S_2)$ with $S_1 = \{C_i\}$ and $S_2 = \{C_j\}$ such that C_i and C_j both have the same sign s_{all} and an intensional definition that subsumes all possible examples. In a such scenario, any example would be immediately be named s_{all} by both agents which is, by definition, a mutual intelligibility. The second scenario would be one agent A_1 copying the current contrast set $K_2 = (U_2, S_2)$ of the other agent A_2. In this situation, the agents have the same contrast set $(K_1 = K_2)$ and therefore have reached mutual intelligibility over any context that is a subset of U_2.

These in these two scenarios, mutual intelligibility has been reached over large contexts. However, if we assume that the initial contrast sets of the agents are a classification that had a purpose, we wish to have any current contrast set also able to fit this purpose no matter no matter what it is. For this reason, the agents only have current contrast sets that are *refinement* of their initial contrast set. By making sure that its new contrast sets are refinements of its initial one, an agent has the guarantee that two examples initially classified in different concept remains labelled as belonging to different classes. This monotonic evolution of contrast sets is formalized in Definition 26.

Definition 26 (Monotonicity)**.** *Let A be an agent that has an initial contrast set $K^0 = (U_0, S_0)$. If A creates a new contrast set $K_1 = (U_1, S_1)$ as its current contrast set, A's contrast sets are* *monotone if* $U_0 \subseteq U_1$*, and if for all pairs of examples* $e_1, e_2 \in U_0$ *and for every concept* $C_1 \in S_1$*, there exists* $C_0 \in S_0$*:*

$$e_1, e_2 \in E(C_1) \Rightarrow e_1, e_2 \in E(C_0)$$

4.5.2 Agreement over Meaning

The mutual intelligibility and the monotonicity are the formalization of the agents goal during the argumentation. In order to reach this goal, the agents need to be able to evaluate them and identify any eventual problem that would prevent the goal's realisation.

Synchronic Agreement

The synchronic agreement is a state of one agent where this agent cannot find any overall pairing relation that would contradict the mutual intelligibility as defined in Conjecture 4. When both agents are in a state of synchronic agreement, we say that the agents have reached mutual intelligibility.

Unlike the mutual intelligibility, the synchronic agreement can be unilateral. This occurs when one agent has access to more overall relations than the other. In this case, the former agent can know about the situation of a pair of concepts that does not satisfy the criteria listed in Definition 25 as presented in Conjecture 4, while the latter agent ignores the situation of this pair of concepts.

Diachronic Agreement

The diachronic agreement is a state of one agent where this agent current contrast set is a refinement of this agent initial contrast set as defined in Definition 26. Unlike the synchronic agreement, the diachronic agreement is always verified. Since the monotonicity is a constraint put on the creation of new contrast sets, no current contrast set can be created in a non-monotonic way.

4.5.3 Types of Disagreements

Synchronic Disagreements

The Conjecture 4 gives, for a list of pairs of concepts and a context, the pairing relations and the relations between the signs that the two concept of each pair should observe in order to have a mutual intelligibility between the agents. If one pair of concept does not observe these properties in the context of the expected mutual intelligibility, the agents do not have the mutual intelligibility. We call such a pair a synchronic *disagreement*. The notion of synchronic disagreement is defined below:

Definition 27 (Synchronic Disagreement)**.** *Let* A_1 *and* A_2 *be two agents that have for respective contrast sets* $K_1 = (U_1, S_1)$ *and* $K_2 = (U_2, S_2)$*. Let* U *be a context such that* $U \subseteq U_1 \cup U_2$*. Let* C_1 *and* C_2 *be two concepts such that* $C_1 \in S_1$ *and* $C_2 \in S_2$*.* A_1 *and* A_2 *have a synchronic disagreement over* C_1 *and* C_2 *within context* U *whenever one of the following conditions holds:*

1. $C_i \odot_U C_j$
2. $C_i \otimes_U C_j$
3. $C_i \dagger_U C_j$

4. $C_i \equiv_U C_j$ *and* $s_i \neq s_j$

5. $C_i \oslash_U C_j$ *and* $s_i = s_j$

or if there exists a concept $C_k \in S_k$ *while there is no concept* $C_{-k} \in S_{-k}$ *such that* $C_k r_U C_{-k}$.

These six conditions give rise to 6 types of disagreement, defined as follows:

Hypo-hypernymy Disagreement If $C_i \odot_U C_j$, then one concept is the hyponym of the other (and the second concept is the hypernym of the first). More specifically, if $r(C_i, C_j, U) = (1, 1, 0)$ then C_i is the *hypernym* of C_j, while if $r(C_i, C_j, U) = (0, 1, 1)$ C_i is the *hyponym* of C_j. A hypo-hypernymy disagreement is expressed as $(s_i, s_j, C_i \odot_U C_j)$.

Overlap Disagreement If $C_i \otimes_U C_j$, the two concepts are said to overlap. An overlap disagreement is expressed as $(s_i, s_j, C_i \otimes_U C_j)$.

Synonymy Disagreement If $C_i \equiv_U C_j$ and $s_i \neq s_j$, we have two concepts that are equivalent but their corresponding signs are different (therefore they are synonyms). A synonymy disagreement is expressed as $(s_i, s_j, C_i \equiv_U C_j)$.

Homonymy Disagreement If $C_i \oslash_U C_j$ and $s_i = s_j$, we have two concepts that are disjoint but their corresponding signs are equal (therefore they are homonyms). A homonymy disagreement is expressed as $(s_i, s_j, C_i \oslash_U C_j)$.

Untranslatability Disagreement If $C_i \not\equiv_U \bullet$, we have a concept C_i that cannot found a concept C_j such that $C_i \equiv_U C_j$. The symbol "$\not\equiv$" does not refer to a paring relation here, but to the absence of a pairing relation of equivalence. Moreover, the symbol "$\bullet$" does not represent a specific concept, but any concept from S_2.

Each of the five first disagreement types can be represented as a triplet (s_1, s_2, r_U) where s_1 and s_2 are the signs of the first and second concepts, and where r_U their relation in the context of the expected mutual intelligibility. Since one type of disagreement corresponds to exactly one type of pairing relation, r qualifies the type of disagreement as it already qualifies the type of pairing relation. The untranslatability disagreement is a special case, noted $(s_1, \bullet, \not\equiv_U)$.

Indistinguishable Disagreement If $C_i \dagger_U C_j$, or if C_i and C_j do not have a pairing relation, the two concepts are said to be indistinguishable. While this disagreement cannot appear with regular (Boolean) r-triplets, it appears the error-tolerant model that requires integer r-triplets in Section 6.1.

Classification of Synchronic Disagreements

Other than according to their types, the synchronic disagreements can be regrouped by families. Synchronic disagreements are regrouped in main families that will later determine the approach that our agents will display to solve them. There are four families of synchronic disagreements: *self*-disagreements, *semantic* disagreement, *lexical* disagreements and *untranslatable* disagreements.

Semantic Disagreement The semantic disagreements are hypo/hypernymy, overlap and indistinguishable disagreements involving two concepts from different agents, that require the refinement of one or two of the agents' concept(s) in order to be solved. Semantic disagreements require either a specific argumentation on meaning in order to create new concepts that are hyponyms of the older ones that cause the disagreement, or the deletion of one of the two concepts in the case of the indistinguishable type of disagreements.

Lexical Disagreement The lexical disagreements are synonymy and homonymy disagreements involving two concepts from different agents, that require a sign change for one or more of the agents' concepts. Lexical disagreements are solved through the creation of new signs, without modifying any other type of semiotic elements.

Self-Disagreement The self-disagreements can be any type of disagreements. However, unlike other families, the self-disagreements involve two concepts that are from the same agent. Due to the fact that the two concepts belong to the same initial contrast-set, which is a partition of the agent's context, a self-disagreement cannot be anything else than an overlap disagreement. Unlike overlaps that belong to the semantic disagreements, the self-disagreements are solved through a process named "border-alignment" instead of creating a new concept.

Untranslatable Disagreements The untranslatable disagreements regroup the disagreements of the eponymous type. With the self-disagreement, this is the only family that does not involve two concepts from different agents, although it is because it only involves one concept. An untranslatable disagreement is solved by creating an equivalent concept to the one involve in the disagreement, and adding it to the contrast-set that misses it.

Diachronic Disagreements

As explained in Section 4.5.2, there is no diachronic disagreement. The fact that an agent A has knowledge over its initial contrast set allows A to create new concepts for its current contrast set that are not violating the diachronic agreement. The fact that the monotonicity is defined on the context of the initial contrast set, that does not change through the argumentation, ensures that no diachronic disagreement can appear following the addition of a new example to the current contrast set's context.

4.5.4 Connected Sets of Disagreements

Our approach is centered on the ability to simplify a communication issue between two agents, involving multiple concepts, into a list of smaller disagreements that each involves only two concepts. At an intermediary level between the interconnected graph of pairing relations and the pairs of disagreement, we have connected sets of disagreements.

Definition 28 (Connected Sets of Disagreements)**.** *Let $D = d_1, \ldots, d_n$ be a set of synchronic disagreements in a context U. Let D_1 be a set of disagreement such that $D_1 \subseteq D$. D_1 is a connected set of disagreement from D if:*

- $\forall d_x \in D_1$ *such that* $d_x = (s_1, s_2, C_1 r_U C_2)$, $\exists d_y \in D_1$ *such that* $d_y = (s'_1, s'_2, C'_1 r_U C'_2)$ *and* $C_1 = C'_1$, $C_1 = C'_2$, $C_2 = C'_1$ *or* $C_2 = C'_2$
- $\forall d_x \in D_1$ *such that* $d_x = (s_1, s_2, C_1 r_U C_2)$, $\nexists d_z \in \{D - D_1\}$ *such that* $d_z = (s''_1, s''_2, C''_1 r_U C''_2)$ *and* $C_1 = C''_1$, $C_1 = C''_2$, $C_2 = C''_1$ *or* $C_2 = C''_2$

Intuitively, connected sets of disagreements are disjoint subsets of a general set of disagreements – usually the set of all the synchronic disagreements between two agents – that are clustered according to the concepts that are causing the disagreements within them.

4.6 Complement to the Notation

4.6.1 On Concepts Sharing Signs

The protocol that we presented in the past sections has been tested in experimental scenarios of increasing complexity. All the scenarios are based on a data set that has been modified in order to create controlled instances of disagreements. Since sometimes the concepts C_i^k from the contrast set K_k of an agent A_k and C_j^{-k} from the contrast set K_{-k} of an agent A_{-k} will be in a situation where $i = j$. In this situation, the concept C_i from K_K and the concept C_j from the hypothesis H_k of A_k can be both noted C_i^k or C_j^k. In order to remove this ambiguity, we will note $C_{j'}^k$ the concept that belongs to H_k. This way, the apostrophe marks the belonging to a hypothesis. In the situation where $i \neq j$, the absence of ambiguity allows us to not put the apostrophe.

4.6.2 On Hyponyms and Hypernyms

During the previous sections, four notions from linguistics (hyponymy, hypernymy, synonymy and homonymy) have been used to characterize the relation between two concepts. We add now a fifth notion, the co-hyponymy. A set of concepts $C_{h_1}, \ldots, C_{h_n}$ are co-hyponyms of a common hypernym C_H if the co-hyponyms' extensional definitions $E_{h_1}, \ldots, E_{h_n}$ are a partition of the hypernym's extensional definition E_H. The correct syntax to express co-hyponymy is: C_{h_1} is co-hyponym *of* C_H *with* $C_{h_2}, \ldots, C_{h_n}$.

Definition 29 (Co-Hyponyms)**.** *Let C_H be a concept, $C_1, \ldots, C_n$ be n concepts with $n > 2$, and U a context such that:*

1. $\forall x \in \{1, \ldots, n\}, Adj(C_x, U) \subset Adj(C_H, U)$
2. $\forall x, y \in \{1, \ldots, n\}, C_x \odot_U C_y$
3. $Adj(C_1, U) \cup \ldots \cup Adj(C_n, U) = Adj(C_H, U)$

then $C_1, \ldots, C_n$ are co-hyponyms of C_H

4.6.3 Computing Multiple R-Triplets and Pairing Relations

When an agent computes multiple r-triplet or pairing relations, we simplify the notation of the set of elements computed. Given two sets of concepts $S_1 = \{C_{1,1}, \ldots, C_{1,m}\}$ and $S_2 = \{C_{2,1}, \ldots, C_{2,n}\}$, the set of R-Triplets $T(S_1 \times S_2, U)$ is equal to:

$$\begin{gathered}\{r(C_{1,1}, C_{2,1}, U),\ \ldots,\ r(C_{1,1}, C_{2,n}, U),\\ \ldots\\ r(C_{1,m}, C_{2,1}, U),\ \ldots,\ r(C_{1,m}, C_{2,n}, U)\}\end{gathered}$$

and the set of pairing relations $R(S_1 \times S_2, U)$ is equal to:

$$\begin{gathered}\{C_{1,1} r_u C_{2,1},\ \ldots,\ C_{1,1} r_u C_{2,n},\\ \ldots\\ C_{1,m} r_u C_{2,1},\ \ldots,\ C_{1,m} r_u C_{2,n}\}\end{gathered}$$

4.7 Conclusion

The formalism presented in this section gives an argumentation model of concepts based on the semiotic triangle. We model the idea of a contrast set as a partition of a context where each part is associated to a specific concept. We introduce the notion of adjunct sets as the relation between the intensional definition of a concept and a particular context. Moreover, we define containers incorporating contrast sets and hypotheses which allows an agent to compare and analyse its contrast set with intensional definitions of the other agent. From here, we defined a typology of pairing relations that characterizes how pairs of concepts relate to each other. We then define r-triplets as the relevant information to characterize a pairing relation between two concepts. Local r-triplets, relating to the context of one agent, are exchanged and the agents can infer the overall pairing relations that hold. This inference holds in the error-free model presented in this chapter; the error-tolerant model, that is built upon this one, is explained in Chapter 6. Finally, we defined mutual intelligibility from the notion of agreement (absence of disagreements).

Chapter 5

Agent Argumentation Model

5.1 A Two Agents Model

All the models and strategies that we explore later in the experiments show similar features. This is due to the fact that each of these models and strategies derives from a unique model that features two agents. These two agents are defined by a set of functions that they both share, by some knowledge that they only partially share and by a set of states which they can go in, each state deciding the agents' actions during the argumentation.

The different models of argumentation on the meaning varies by the error management of the model's machine learning component. On the other hand, the different strategies of argumentation on the meaning varies by the set of states in which the agents can go in.

We qualify our model as symmetrical, since for a given model and strategy the set of functions of the agents is identical, the set of states of the agents is identical, and the knowledge shared by one agent is also shared by the other.

5.1.1 Agent Knowledge

While we are in the position of an oracle and have access to the entire knowledge of both agents, each agent has itself limited knowledge over the elements that are belonging to the other agent.

In order to clarify to which knowledge which agent has access to, we are classifying the agents' knowledge in three categories: *personal* knowledge, *overall* knowledge and *inferred* knowledge.

The personal knowledge is only accessible to one agent A_k. The overall knowledge is accessible to both agents A_k and A_{-k}. The inferred knowledge is a knowledge accessible to A_k that is mirroring, with a varying degree of accuracy, a knowledge either only accessible to A_{-k} or a knowledge accessible neither to A_k or A_{-k}.

Personal Knowledge

An agent A_k has access to a contrast set, called the initial contrast set K_k^0, and all the elements from this contrast set – the contrast set's concepts and the semiotic elements from these concepts. Agents can also create new contrast sets. If an agent comes to create a new contrast set, called the *current* contrast set K_k by opposition to the initial one, this agent has also access to this contrast set. When we mention a contrast set without precision on whether it is the initial or current contrast set, it is by default the current contrast set.

Overall Knowledge

The overall knowledge is the knowledge that one agent shares, at least partially, with the other agent. This category of knowledge can be divided in two categories: the *shared* knowledge that is directly shared with the other agent through messages, and the *inferred* knowledge that is deduced from the other agent's knowledge without being directly exchanged by messages.

Shared Knowledge The shared knowledge is transmitted through messages. Since a message can only contain either semiotic elements or triplets, the shared knowledge is limited to some semiotic elements from both agents and their relations. Since exchanging messages has a cost, the agents try to reduce the amount of shared knowledge during an argumentation. The examples that an agent has sent or received are recalled in a specific set, written U_{ex}. This set of example helps the agent to not exchange an example twice, and is used to infer additional knowledge. The generalizations, always exchanged by sets corresponding to intensional definitions, can be stored in an agent A_k's hypothesis H_k when they are received. The intensional definitions are always received with an associated sign. While the hypothesis of an agent contains concepts and not intensional definitions alone, only the intensional definitions of the hypotheses' concepts and their signs are shared by both agents. The extensional definitions from the hypotheses' concepts are only inferred by the agent. The triplets received are used to infer overall triplets and overall pairing relations, which is discussed in the *inferred knowledge* paragraph.

Inferred Knowledge From the intensional definitions received by an agent, the other agent can infer an approximation of the other agent's concept associated extensional definition (see Section 5.1.3) by using its own adjunct set instead. This allows the agent to create a new concept that can be stored in its hypothesis, which represents the agent's guess on the concepts of the other agent. An agent can infer the overall relation between one of its concept and one of the other agent's concept as described in Section 4.4.3. Once an overall pairing relation has been inferred, this agent can identify if this pairing relation induces a disagreement. If the paring relation is causing a disagreement, this disagreement is formalized into a triplet (see Section 4.5.3) and recalled in a set of active disagreements D.

The distinction between personal knowledge and inferred knowledge is perfectly illustrated in the Section 5.2 with the beliefs and arguments. While an e-argument represents a knowledge that is known as a fact by an agent, a belief represents a knowledge that the agent has inferred but is partially unsure of. Personal knowledge is exchanged by an agent to attack the inaccurate parts of the inferred knowledge of another agent. On the other hand, shared knowledge is supposed to be known by both agents and therefore should not be exchanged at all.

5.1.2 Agent Modus Operandi

The argumentation takes place turn by turn, with one agent receiving a token at the beginning of the argumentation, taking a set of actions, then passing the token to the other agent. An agent can take actions only when it has the token, and the last action that it takes is always to pass the token to the other agent. The token is exchanged until termination is detected. The actions taken by an agent while it has the token are decided by the agent's inputs and its current state. The two variables that impact the behaviour of one agent at a given turn are the agent's state (a qualitative variable) and the messages that this agent has received from the other agent. Each agent has the same set of possible states, making our argumentation model *symmetric*.

The argumentation model is also synchronized, meaning that if an agent A_1 is in a state #1 during the turn t, the agent A_2 was either in the state #1 during the previous turn t_{-1} or will be in the state #1 during the next turn t_{+1}.

5.1.3 Agent Functions

Inductive Learning

The agents are inductive learners. Being able to use inductive learning over a set of examples in order to obtain a set of generalizations that subsume these examples without subsuming the rest of the subset is the most fundamental function of our agents. Each agent use the ABUI algorithm in order to achieve inductive learning. An agent A with a local context U_k needs to split its context in two subsets E and E' such that $E \cup E' = U_k$ and $E \cap E' = \emptyset$. Once these two sets of examples have been created and passed as inputs of ABUI, ABUI returns an intensional definition $I = \{g_1, \dots, g_n\}$ such that:

- $\forall e \in E, \exists g \in I$ such that $g \sqsubseteq e$
- $\forall e \in E', \nexists g \in I$ such that $g \sqsubseteq e$

The ABUI algorithm cannot always guarantee that the intensional definition returned verifies these two properties, but it approaches them as much as it can by minimizing the number of examples from E' that are subsumed by I, and the number of examples from E that are not subsumed by I. While the rest of this section does not take into account these errors of first and second order, their impact is discussed later in Section 6.1.

Naming Examples

The agents can name the examples presented to them. When an agent names an example e, it always uses a left-path associations to find e's associated sign. The container used by the agent to name e is its current contrast set K, or the initial contrast set if no current contrast set had been created by the agent. When naming an example e with its contrast set K, an agent returns the set of signs $\{s_1, ..., s_n\}$ such that $e \underset{K}{\overset{l}{\mapsto}} \{s_1, ..., s_n\}$.

Sending and Receiving Messages

An agent can send a message to another agent. A message has two parts: a performative and a content. The performative of the message indicates to the other agent the intent of the message, while the content vehicles a knowledge that the sender has. The different types of messages are presented in Appendix B along with their associated performatives and contents.

When an agent sends a message, it arrives in the mailbox of the other agent. The message stays in the mailbox until the other agent removes it. In Section 5.1.2, we mentioned that agents were taking actions turn by turn. These actions are mostly determined by the state of an agent during its turn, but also by the messages that are in the mailbox. It can happen that a message from an agent A_1 is not supposed to be read by an agent A_2 before A_2 is a certain state. For this reason, each message is timed for a defined state written #State. The message will only be delivered to the other agent when this agent enters the timed state for the next time.

For instance, in our model, the agents A_1 and A_2 goes through two states #1 and #2 where they respectively evaluate the local r-triplets of each other, and the overall r-triplets of each other. Let's assume that A_1 enters State #1 first. It is supposed to have received (local) r-triplets from *Evaluation* messages sent by A_2, and to compare these r-triplets with its own local r-triplets. Doing so allows A_1 to find the overall r-triplets, as explained in 4.4.3. After this, A_1 sends the overall r-triplets that it just found to A_2 with an *Evaluation* message, and passes the token to A_2. Upon receiving the token, A_2 enters State #1 and follows the same instructions as A_1 did during the previous turn. However, additionally to the *Evaluation* messages containing the local r-triplets from A_1 that any agent is supposed to have received at the beginning of State #1, A_2 has also received the overall r-triplets sent by A_1 during the previous turn, and that are not needed until State #2. Since both sets of messages carry the same performative and the same type of content, A_2 does not know which r-triplet it is supposed to consider as local r-triplets to compute the overall r-triplets, and which r-triplets it is supposed to ignored. Timing the *Evaluation* messages sent by A_1 during State #1 in order for them to be received by A_2 only when A_2 enters State #2 solves this issue.

The notation of a message is the following; a message with:

- a performative *Performative*,
- a content $T = \{T_1, \ldots, Tn\}$ were each T_i is a different type of content, and
- timed for the agent state #State

is written *Performative*#State(T_1, ..., T_n). Certain elements of a message have identifiers. If an agent A_1 is about to send an element that has an identifier, and knows that the other agent A_2 already has access to an element with the same identifier, the agent A_1 automatically substitutes this element by its identifier.

Computing Adjunct Sets

After naming examples, the most basic function that an agent should have is to compute adjunct sets of concepts. This function links the left-path and the right-path associations by retrieving the set of examples subsumed by the intensional definition of a concept. By default, the adjunct set is computed using the current contrast set of the agent. The notion of adjunct set is defined in Definition 17. An agent A_k computes the adjunct set of a concept $C = \langle s, I, E\rangle$ such that $I = \{g_1, \ldots, g_n\}$ by directly creating the set of examples $Adj(C, U_k) = \{e \in U_k | \exists g \in I \wedge g \sqsubseteq e\}$.

Computing Local R-Triplet and Pairing Relation

Using its adjunct sets, an agent A_k can find the local pairing relation between two concepts C_i and C_j as explained in Section 4.4.1. First, A_k builds the local r-triplet $r(C_i, C_j, U_k)$, and then computes the local pairing relation r_{Uk} using Definition 22. The computation of a local R-Triplet and the computation of its associated relation has already been detailed in Figure 5.1. In this figure, two adjunct sets are isolated from one context Q which contains eight examples. The two adjunct sets are used to compute three partial sets a, b and c. The size of these pairing partial sets are used to determine the local r-triplet of the relation, which is then translated as a pairing relation type.

Inferring Overall Pairing Relation from Received Triplets

Using the composition law presented in Conjecture 2, an agent can infer an overall pairing relation between two concepts C_i and C_j from a local pairing relation received from another agent and its own corresponding local pairing relation. However, the two agents need the intensional definitions of both C_i and C_j in order to compute their local r-triplets and infer the overall pairing relation. Once the overall relation of two concepts is obtained, the agents know if these two concepts are causing a semantic or lexical disagreement.

In an error-free model, having both local triplets is enough to find the overall relation between two concepts from different agents. However, in an error-tolerant model, the agents might need to

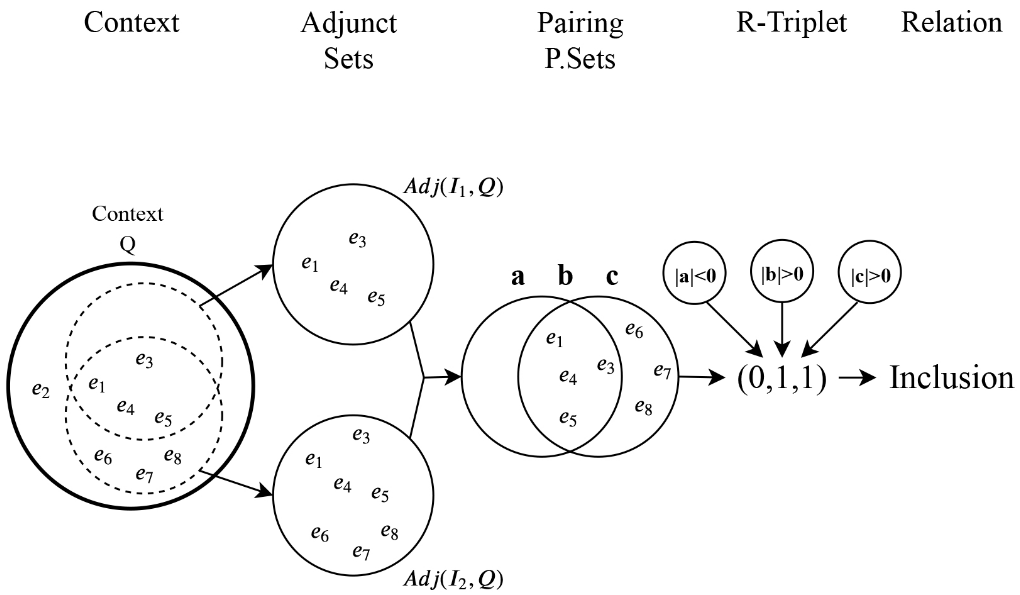


Figure 5.1: Relation between the context, concepts' adjunct sets, pairing partial sets, r-triplets and pairing relations. The left-path (intensional definitions) partitions a context into adjunct sets. Two adjunct sets create three theoretical pairing partial sets. The count of examples in each pairing partial sets gives the r-triplet of the relation between the two adjunct sets. The r-triplet gives the quality of the pairing relation of the concepts from with the left-paths have been borrowed to make adjunct sets.

exchange more knowledge. This problem is discussed in Section 6.1. The computation of the overall pairing relation from two local ones is represented in Figure 5.2. In this figure, the agents compute different local r-triplets that are conflicting with each others, one resulting in a disjunction pairing relation and the other in an inclusion pairing relation. However, combining them with Conjecture 2 results in an overall r-triplet corresponding to the overall r-triplet of the two Concepts C_1 and C_2.

Disagreement Listing

Finding and listing the disagreements in order to resolve them is one of the main functions of the agents. In order to resolve a disagreement, both agents should be aware of its existence and have characterized it: they should know which signs and pairing relation (or absence of relation) is behind it. That is why the disagreements are always characterized in the overall context. Before starting to resolve their disagreements, the agents should exchange enough knowledge to be certain of the nature of the eventual overall relations that cause these disagreements.

As soon as two local relations involving the two same concepts are exchanged, the agents can categorize and list the semantic, lexical or self disagreements depending on the inferred overall relation and origin of the two concepts. An untranslatable disagreement, on the other hand can only be listed after an extensive transfer of the agents' intensional definitions and the inference of all the overall pairing relations, as it is characterized by the *absence* of a pairing relation.

In the case of the lazy strategy presented in Chapter 8, the agents do not exchange all their intensional definitions at the start of the argumentation, and therefore have to be vigilant on the fact that each concept belonging to a same system of disagreement should have an equivalent in another contrast-set.

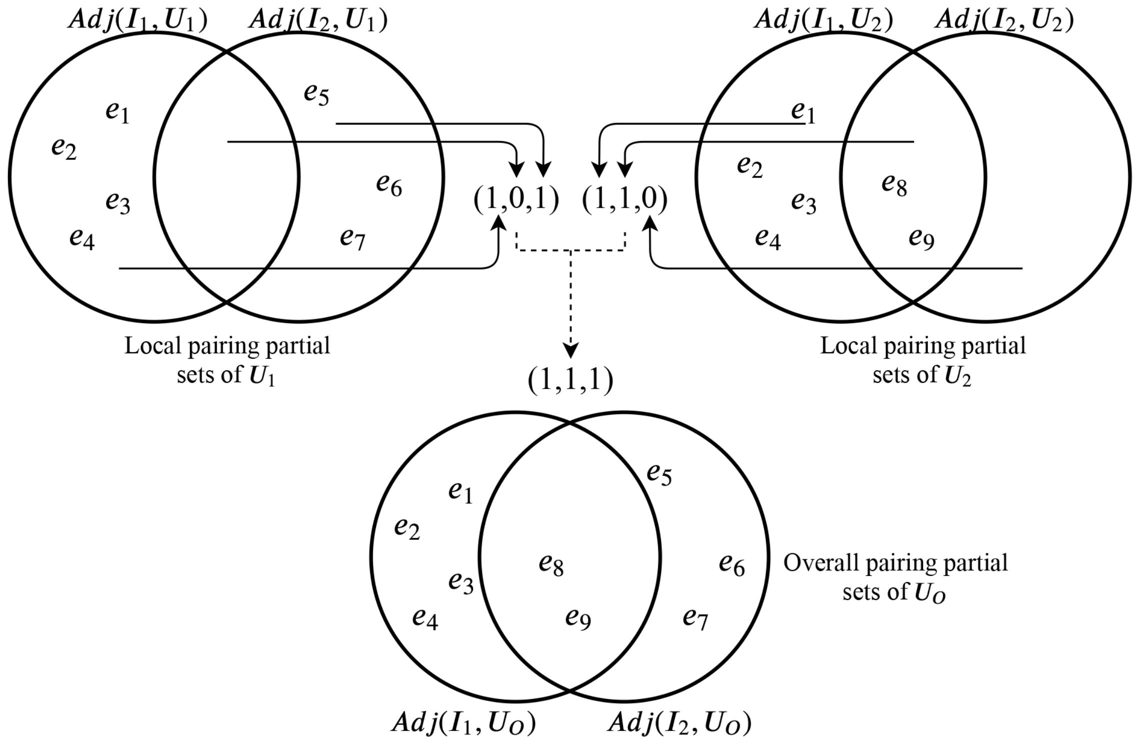


Figure 5.2: Example of overall pairing relation computation.

Creating New Signs

Each agent can create a sign. The creation of a new sign is simple, as the signs are in an arbitrary relation with the two other semiotic elements. The created signs are ensured to be all different by keeping track of the previously created signs, and by choosing a specific structure for the new signs that differentiates them from the signs that are already existing in the agents' contrast sets. For instance, in our implementation, all the new signs start with the radical *newSign_*, followed by a unique number that is incremented each time that a sign is created.

The sign created by an agent can also be used by another agent, as long as this sign has been sent to the other agent along with the semiotic element to which it is associated.

Creating New Concepts from Right-Path Associations

An agent can learn a new concept if it received a set of examples associated to one sign. If the agent receives the class $U(\mapsto s)$, it can learn by inductive learning a new intensional definition $I = \{g_1, \ldots, g_n\}$ such that I subsumes U. Other classes can be involved in the learning as negative examples: for instance, if an agent receives two classes $U_+(\mapsto +)$ and $U_-(\mapsto -)$ and wishes to create a concept that corresponds to the first class, it can learn an intensional definition I_+ that covers the examples U_+ without covering the examples U_-.

In our model, these intensional definitions are learned through inductive learning, using the ABUI algorithm. Any machine learning technique can be used to create a set of generalizations that respect these properties (only subsuming the examples from the designed class). However, in the case where an absolutely accurate learning is not guaranteed or possible, the model needs to be adapted to account a certain degree of error. The aforementioned adaptations are presented in Section 6.1.

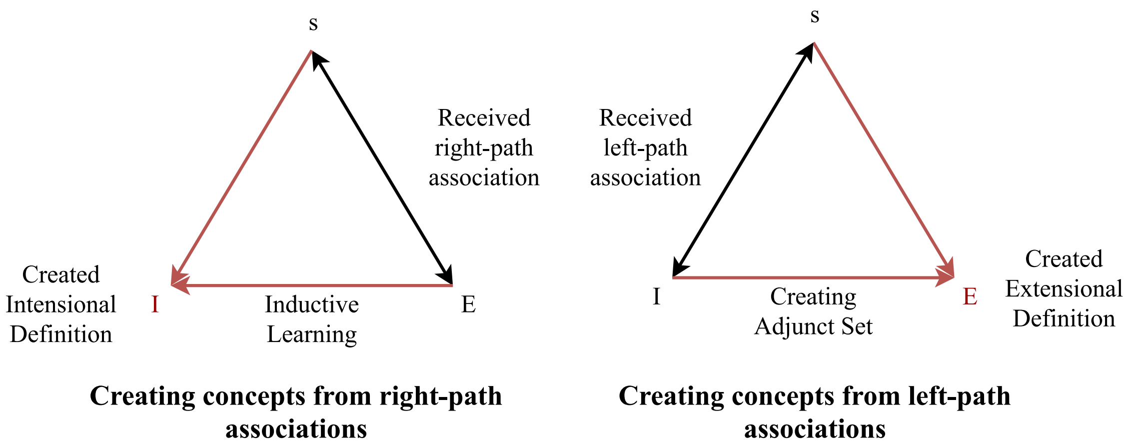


Figure 5.3: The creation of a concept $C = \langle s, I, E\rangle$ from a right-path association (left) and a left-path association (right), by respectively retrieving the intensional definition through inductive learning and retrieving the extensional definition through the computation of the adjunct set.

Once the agent has created an intensional definition I, it creates the new concept $C = \langle s, I, U_+\rangle$ as the association of the three semiotic elements. Agents mostly create concepts in this fashion to create their initial contrast sets, when they receive their initial sets of right-path associations. They also use this method to generate new beliefs, when they need to cooperatively create new concepts with other agents without having an intensional definition upon which to base this creation.

Creating New Concepts from Left-Path Associations

When an agent does not directly have access to the right-path associations, it can still create a concept from an intensional definition I and a sign s by creating the adjunct set of the concept from which I and s originates, as the adjunct set only requires an intensional definition to be computed. By doing so, the agent obtains a set of left-path associations. If an agent A_k receives an association $I = \{g_1, \dots, g_n\} \mapsto s$, it can directly create the concept $C = \langle s, I, Adj(I, U_k)\rangle$.

Creating New Concepts through Argumentation

Creating a new concept $C_n = \langle s_n, I_n, E_n\rangle$ using either left or right-path associations requires for the agent to have at least two of C_n's semiotic elements – its sign and either its extensional or intensional definition. However, our model requires the agents to create new concepts which they have no semiotic element of, in order to resolve several types of disagreements. In situations like these, the agents can only create a new concept by arguing with each others.

First, the two agents need to determine which subset of the overall context will be the extensional definition of a new concept. This set is written $U_n^+ = Adj(C_n, U_O)$, as the extensional definition of our new concept C_n should ideally be C_n's adjunct set according to Definition 17. In the context of a disagreement which involves two concepts C_1 and C_2, U_n^+ is determined to be one of the overall pairing partial sets $U_{O,1,\overline{2}}, U_{O,1,2}, U_{O,\overline{1},2}$ of C_1 and C_2. The choice of a particular set depends on the type disagreement that the agents are resolving, which we discuss later in Section 5.5.

The relative complement of U_n^+ with respect to U_O is written U_n^-. Together, U_n^+ and U_n^- partition the overall context U_O. Even if it has determined which overall pairing partial set of the pair of concept C_1, C_2 is U_n^+, A_k cannot directly access the sets U_n^+ and U_n^- as it might contains examples from U_{-k}. A_k can only access its local examples of U_n^+, the set $U_{n,k}^+ = U_n^+ \cap U_k$.

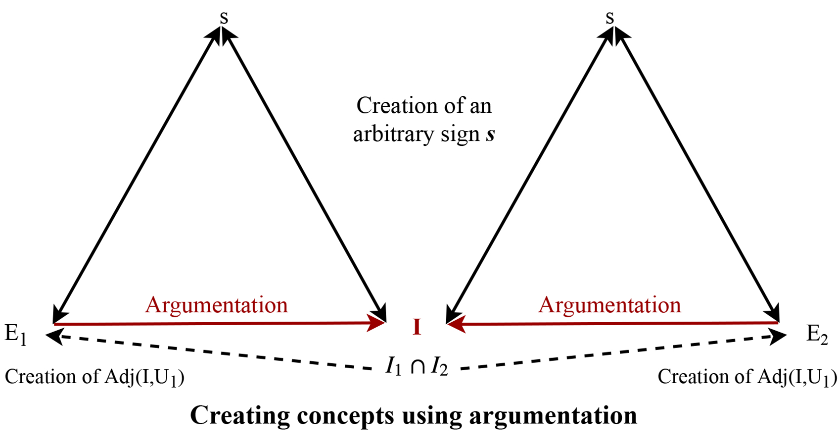


Figure 5.4: The creation of a new concept through the use of argumentation. The main steps are: determining an overall pairing partial sets from two intensional definitions, creating the corresponding local partial sets, arguing on the adequacy of proposed intensional definitions for the overall pairing partial set, and creating an arbitrary sign.

The next step for the agents is to create an arbitrary sign s_n that each agent A_k will associate with its set of examples $U^{+}_{n,k}$. Since s_n only has the requirement of not belonging to the agents' joint vocabulary, any agent can create s_n and send it to the other agent. Once both agents have associated their sets of examples $U^{+}_{n,k}$ with the sign s_n, they each have a set of association $U^{+}_{n,k} \overset{r}{\underset{k}{\mapsto}} s_n$. One agent A_1 is then chosen to create a new intensional definition from this set of right-path associations, using the same method as for the creation of new concepts using right-path associations detailed in Section 5.1.3. A_1 is chosen accordingly to the type of disagreement that C_n is supposed to help resolving, which we discuss later in Section 5.5.

Since $U^{+}_{n,1}$ is only a subset of U^{+}_{n}, there is an important risk that the intensional definition I^{t1}_{1} which is learned over $U^{+}_{n,1}$ subsumes examples subsumes examples that belong to the set $U^{-}_{n,2} = U^{-}_{n} \cap U_2$, or on the contrary does *not* subsume examples from $U^{+}_{n,2}$. Since $U^{+}_{n,2}$ and $U^{-}_{n,2}$ are also subsets of respectively U^{+}_{n} and U^{-}_{n}, a such scenario would imply that I^{t}_{n} is not fit to be the extensional definition of U^{+}_{n}. In order to help A_1 to create a suitable intensional definition for U^{+}_{n}, the agent A_2 can argue about the fitness of I^{t1}_{n} over U^{+}_{n} so A_1 can create a more fitting intensional definition I^{t2}_{n}. This argumentation can be done over any intensional definition I^{tx}_{n} until a final intensional definition I^{tF}_{n} that is suitable for U^{+}_{n} is found. The argumentation model used to find I^{tF}_{n} is discussed in Section 5.1.3. Once the intensional definition is found, each agent A_k creates its own version of the new concept $C^{k}_{n} = \langle s_n, I^{tF}_{n}, U^{+}_{n,k} \rangle$. Then, A_1 adds C^{1}_{n} to its contrast set while A_2 adds C^{2}_{n} to its hypothesis.

Managing the Creation of a New Intensional Definition

As mentioned in the previous section, only one agent is in charge of a new concept's creation through argumentation while the other agent is helping by arguing over the correctness of the relation between the new concept's extensional and intensional definitions. When an agent learns that it will have to create a new concept through argumentation, it decides whether or not to take the lead according to the following rules. The agent decides to take the lead according to the protocol described in Section 5.5.5.

5.2 Argumentation in Concept Creation

5.2.1 Intensional Definitions as Binary Classifications

Before arguing in the process of a new concept C_n's creation, the agents have already determined the extensional definition and the sign of the new concept. The agents are however missing the intensional definition $I(C_n)$, and the argumentation is what will help them to find it. As we mentioned in Section 5.1.3, the main issue with finding $I(C_n)$ is that the agents need to ensure that $I(C_n)$ is subsuming every elements of U_n^+ without subsuming any element of U_n^-. This requires to test all the elements of the overall context, which cannot be done by one agent individually as none of the agents is supposed to have access to the overall context.

The agents can however overcome this issue by cooperating in the creation of $I(C_n)$. In order to explain how, let's consider the creation of $I(C_n)$ as a binary classification over U_O. The agents want $I(C_n)$ to subsume the examples of the extensional definition U_n^+ without subsuming the other examples from U_O, U_n^+ is similar to a set of positive elements. Likewise, the set of examples U_n^- is similar to a set of negative elements since it is the relative complement of U_n^+ with regard to the entire population of our test U_O. By subsuming or not the examples of U_O, $I(C_n)$ creates another partition of U_O into positive and negative assignments. The positive assignments of $I(C_n)$ are the examples $\{e \in U_O | I(C_n) \sqsubseteq e\}$, which according to Definition 18 is the adjunct set of $I(C_n)$. The set of negative assignments of $I(C_n)$ is $\{e \in U_O | I(C_n) \not\sqsubseteq e\}$, equivalent $U_O - Adj(I(C_n), U_O)$. In order to be an intensional definition for C_n.

According to Definition 11, $I(C_n)$'s set of positive assignments should be equivalent to the set of positive elements in order for $I(C_n)$ to be an adequate intensional definition for C_n. Representing $I(C_n)$ as a binary classification allows us to measure the adequacy of any intensional definition I to the role of intensional definition of C_n in terms of true positives, true negatives, false positives and false negatives. These four sets are defined as:

- The set of true positives of I is $TP(I) = \{e \in U_n^+ | I \sqsubseteq e\}$
- The set of true negatives of I is $TN(I) = \{e \in U_n^- | I \not\sqsubseteq e\}$
- The set of false positives of I is $FP(I) = \{e \in U_n^- | I \sqsubseteq e\}$
- The set of false negatives of I is $FN(I) = \{e \in U_n^+ | I \not\sqsubseteq e\}$

If the intensional definition has I no false positives or false negatives, therefore the set of I's positive assignments is equivalent to the set of positive elements and I is suitable to be the intensional definition $I(C_n)$ of the new concept C_n. Therefore, in order to be sure that $I = I(C_n)$, the agents should agree that both $FP(I)$ and $FN(I)$ are empty. However, as we mentioned, no agent has individually entirely access to U_n^+ or U_n^-. But since the overall context is the union of the two local contexts, we know that:

- $(FP(I) \cap U_1 = \emptyset) \wedge (FP(I) \cap U_2 = \emptyset) \Leftrightarrow FP(I) = \emptyset$
- $(FN(I) \cap U_1 = \emptyset) \wedge (FN(I) \cap U_2 = \emptyset) \Leftrightarrow FN(I) = \emptyset$

While the agents cannot directly access U_n^+ and U_n^-, they can both locally verify that they have no knowledge of false positives or negatives and share this information in order to know if an intensional definition is suitable to be the intensional definition of C_n. Exchanging false positives and negatives is akin to an argumentation process.

5.2.2 Argumentation Model

In formal argumentation theory, an abstract argumentation framework consists of a combination of a set of abstract elements $A = \{\alpha_1, \ldots, \alpha_n\}$ called *arguments*, and a binary relation R on A^2 called *attack* relation. A specificity of our model is that an attack relation is always univalent. An argument α attacking another argument α' is written $\alpha \twoheadrightarrow \alpha'$. In our argumentation model, an argument α represents a binary classification of a context U, partitioning U into positive and negative assignments.

In our argumentation model, each argument can be related to one of the two agents A_1 and A_2 that are arguing, to a set of examples called its *extension*, to a sign called its *label*, and to an intensional definition called its *intension*. An argument can also be *marked* by a marking function.

Definition 30 (Belonging of Argument). *Let $Ag = \{A_1, A_2\}$ be a set of two agents, and $A = \{\alpha_1, \ldots, \alpha_n\}$ a set of arguments. A* belongs to *relation is a univalent binary relation over A and Ag that can relate an argument $\alpha \in A$ to exactly one agent $A_k \in Ag$.*

Definition 31 (Label of Argument). *Let $S = \{s_1, s_2\}$ be a set of two signs, and $A = \{\alpha_1, \ldots, \alpha_n\}$ a set of arguments. A* labels *relation is an injective binary relation over S and A that can relate exactly one sign $s \in S$ to an argument $\alpha \in A$.*

Definition 32 (Extension of Argument). *Let $\mathbb{U}$ be the set of all possible sets of examples, and $A = \{\alpha_1, \ldots, \alpha_n\}$ a set of arguments. The relation* is extension of *is an injective binary relation over $\mathbb{U}$ and A that can relate exactly one set of examples $U \in \mathbb{U}$ to an argument $\alpha \in A$.*

Definition 33 (Intension of Argument). *Let $\mathbb{I}$ be the set of all possible sets of generalizations, and $A = \{\alpha_1, \ldots, \alpha_n\}$ a set of arguments. The relation* is intension of *is an injective binary relation over $\mathbb{I}$ and A that can relate exactly one intensional $I \in \mathbb{I}$ definition to an argument $\alpha \in A$.*

Definition 34 (Marking Function). *Let $A = \{\alpha_1, \ldots, \alpha_n\}$ be a set of arguments and R an attack relation on A^2. The marking function $m : A \times R \rightarrow \{\text{accepted}, \text{rejected}\}$ is a function such that for every argument $\alpha \in A$:*

$$m(\alpha, R) = \begin{cases} \text{rejected}, & \textit{if there exists } \alpha' \twoheadrightarrow \alpha \textit{ in } R, \textit{ and if } m(\alpha', R) = \text{accepted}. \\ \text{accepted}, & \textit{otherwise.} \end{cases} \tag{5.1}$$

In our model, the agents use arguments to cooperatively create new intensional definitions. In order to do so, they have one argumentation *tree* in their argumentation. An argumentation tree is a tuple of sets of agents, signs, arguments and relations, written $\Gamma = \langle Ag, S, A, A', R, B, L, E, F \rangle$, that is defined in Definition 35. The argumentation trees of our model need to respect some properties. An argumentation schema is a step in the process of the creation of a new intensional definition in the context of concept creation through argumentation.

Definition 35 (Argumentation Tree). *Let $Ag = \{A_1 \text{ and } A_2\}$ be a set of two agents, and S the set of signs $\{+, -\}$. Let $A = \{\alpha_1, \ldots, \alpha_n\}$ be a set of arguments called set of* active *arguments and A' a subset of A called set of* accepted *arguments, R an attack relation over A^2, B a* belongs to *relation over Ag and A, L a* labels *relation over A and S, E an* is extension of *relation over A and the set of all possible sets of examples $\mathbb{U}$ and F an* is intension of *relation over the set of all possible sets of generalizations $\mathbb{I}$ and A. The tuple $\Gamma = \langle Ag, S, A, A', R, B, L, E, F \rangle$ is an argumentation tree if:*

- *there is exactly one argument α in A such that α attacks no other argument from A, called* root *of Γ.*

- *for each agent A_k in Ag, there are no arguments α, α' such that (α, A_1) and (α', A_2) exists in B, and $\alpha \twoheadrightarrow \alpha'$ exists in R.*

With the notion of argumentation tree, we can now formally define an argumentation schema. This definition holds in the context of the creation of a new concept, and does not cover a general strategy of argumentation, presented in this thesis.

Definition 36 (Argumentation Schema). *Let A_1 and A_2 be two agents that have U_O for overall context, Γ an argumentation tree that a set of agents $\{A_1, A_2\}$, U^+ a set of examples such that $U^+ \subset U_O$. The tuple $\Delta = \langle A_1, A_2, \Gamma, U^+ \rangle$ is an argumentation schema. We say that Δ is an argumentation schema between A_1 and A_2, that Δ is an argumentation schema over U_O, and that U^+ is the set of positive examples of Δ.*

We use the notation presented in Definition 38 to refer to the specific constituents of an argumentation schema Δ, and the notation presented in 37 to refer to the specific constituents of an argumentation tree Γ.

Definition 37 (Argumentation Tree Constituents). *For any argumentation tree Γ such that $\Gamma = \langle Ag, S, A, A', R, B, L, E, F \rangle$, $Ag(\Gamma)$ is the set of agents Ag, $S(\Gamma)$ is the set of signs S, $A(\Gamma)$ is the set of active arguments A, $A'(\Gamma)$ is the set of accepted arguments A', $R(\Gamma)$ is the* attack *relation R, $B(\Gamma)$ is the* belongs to *relation B, $L(\Gamma)$ is the* labels *relation L, $E(\Gamma)$ is the* is extension of *relation E and $F(\Gamma)$ is the* is intension of *relation F.*

Definition 38 (Argumentation Schema Constituents). *For any argumentation schema Δ equals to $\langle A_1, A_2, \Gamma, U^+ \rangle$, $Ag(\Delta)$ is the set of agents $\{A_1, A_2\}$, $U_+(\Delta)$ is the set of examples U^+, $U_O(\Delta)$ is the overall context of A_1 and A_2 and $\Gamma(\Delta)$ is the argumentation tree Γ.*

Let A_1 and A_2 be two agents, and U^+ and U^- two sets partitioning the overall context $U_O = U_1 \cup U_2$ of A_1 and A_2. The two agents A_1 and A_2 aim to create a new concept C through argumentation, such that C has for adjunct set the set of examples U^+ in the overall context. The agents will create a new argumentation schema $\Delta_0 = \langle A_1, A_2, \Gamma_0, U^+ \rangle$ with an argumentation tree $\Gamma_o = \langle \{A_1, A_2\}, \{+, -\}, A_0, A'_0, R_0, B_0, L_0, E_0, F_0 \rangle$, such that $A_0 = A'_0 = \emptyset$, and the binary relations R_0, B_0, E_0, F_0 do not contain any pair. The agents will then, turn by turn, add new arguments in the argumentation schema Δ_0 that will change both Δ_0 and its argumentation tree, until the agents find after n turns an argumentation schema $\Delta_n = \langle A_1, A_2, \Gamma_n, U^+ \rangle$ such that the root of Γ_n represents an acceptable binary classification for U^+.

5.2.3 Argument Types and their Insertion and Deletion

There are three different types of arguments that can be added to an argumentation schema: the *root*-arguments, the *g*-arguments and the *e*-arguments, and the agents need to follow certain rules specific to each type when inserting or deleting an argument to the argumentation schema. An argument added to an argumentation schema is directly added to its argumentation tree.

Definition 39 (Root-Argument). *Let A_k be an agent, s a sign and U a set of examples and I a set of generalizations. The tuple $\alpha = \langle s, A_k, I \rangle$ is a root-argument.*

Definition 40 (E-Argument). *Let A_k be an agent, α' an argument, s a sign and U a sets of examples. The tuple $\alpha = \langle \alpha', s, A_k, U \rangle$ is an e-argument.*

Definition 41 (G-Argument). *Let A_k be an agent, α' an argument, s a sign, and I a set of generalizations. The tuple $\alpha = \langle \alpha', s, A_k, I \rangle$ is a g-argument.*

Let $\Gamma_t = \langle Ag, S, A_t, A', R_t, B_t, L_t, E_t, F_t \rangle$ be an argumentation schema. In order to insert a new root-argument $\alpha = \langle s, A_k, I \rangle$ in Γ_t, an agent $A_k \in Ag$ must replace Γ_t by a new argumentation tree Γ_{t+1} such that the set of active arguments A_{t+1} is equal to A_t, the set of accepted arguments A'_{t+1} is equal to A'_t, the *attack* relation R_{t+1} is equal to $R_t \cup \alpha \twoheadrightarrow \alpha'$, the *belongs to* relation B_{t+1} is equal to $B_t \cup (\alpha, A_k)$ the *labels* relation L_{t+1} is equal to $L_t \cup (s, \alpha)$, the *is extension of* relation E_{t+1} is equal to E_t, and the
emphis intension of relation F_{t+1} is equal to $F_t \cup (\alpha, I)$.

In order to insert a g-argument $\alpha = \langle \alpha', s, A_k, I \rangle$ to Γ_t, the agent A_k replaces Γ_t with the same new argumentation tree Γ_{t+1} than it would have for a root-argument $\langle s, A_k, U, U' \rangle$, except for the relation A_{t+1} which is then equal to $A_t \cup \alpha \twoheadrightarrow \alpha'$.

In order to insert a e-argument $\alpha = \langle \alpha', s, A_k, U \rangle$ to Γ_t, the agent A_k replaces Γ_t with the same new argumentation tree Γ_{t+1} that it would have for a root-argument $\langle \alpha', s, A_k, U, U' \rangle$, except for the relation E_{t+1} which is then equal to $E_t \cup (U', \alpha)$, and the relation F_{t+1} which is then equal to F_t.

An argument can only be inserted once, which means that R_{t+1}, B_{t+1} and I_{t+1} always remain univalent while L_{t+1} and E_{t+1} always remain injective.

An agent can also delete an argument α from an argumentation schema that has an argumentation tree Γ_t. It does so by replacing Γ_t with another tree Γ_{t+1} where α has been removed from the set of active arguments of Γ_t, and where all pairs from all the relations of Γ_t that contains α have been removed from their corresponding relations in Γ_t.

5.2.4 Classification and Agreement upon Arguments

An agent A_k either agrees or disagrees upon all arguments α of an argumentation schema Δ. Let Γ_t be the argumentation tree of Δ where α belongs to the set of active arguments. In order to determine if it agrees upon α, A_k needs to determine two other elements: the *coverage* $V(\alpha, \Delta)$ of the argument α is assimilated to the set of examples that are positive assignments of α, and the *target* $T(\alpha, \Delta)$ of α which is assimilated to the set of examples that are positive values in the classification attempt that α makes. — classifications have positive and negative values and assignments that happens here to be examples. I added that they were examples to make it more intuitive.

Definition 42 (Coverage)**.** *Let $\mathbb{U}$ be the set of all possible sets of examples and $\mathbb{G}$ the set of all possible argumentation schemas. Let $V : A(\Gamma(\mathbb{G})) \times \mathbb{G} \rightarrow \mathbb{U}$ be a function that, for every argumentation schema $\Delta \in \mathbb{G}$ and every argument α in $A(\Gamma(\Delta))$, yields a set of examples defined as follows: :*

$$V(\alpha, \Delta) = \begin{cases} Adj(I, U_O(\Delta)), & \text{if } (I, \alpha) \in F(\Gamma(\Delta)). \\ U, & \text{if } (U, \alpha) \in E(\Gamma(\Delta)). \end{cases} \tag{5.2}$$

We call the set of examples $V(\alpha, \Delta)$ the coverage *of α in Δ.*

Definition 43 (Target)**.** *Let $\mathbb{U}$ be the set of all possible sets of examples and $\mathbb{G}$ the set of all possible argumentation schemas. The target function $T : A(\Gamma(\mathbb{G})) \times \mathbb{G} \rightarrow \mathbb{U}$ is a function such that for every argument α in $A(\Gamma(\Delta))$ and every argumentation schema $\Delta \in \mathbb{G}$:*

$$T(\alpha,\Delta) = \begin{cases} U_+(\Delta), & (\alpha \twoheadrightarrow \alpha) \notin R(\Gamma(\Delta)). \\ V(\alpha',\Delta) - U_+(\Delta), & (\alpha \twoheadrightarrow \alpha') \in R(\Gamma(\Delta)), \textit{ and } (s,\alpha) \textit{ and } (s',\alpha') \textit{ both exists} \\ & \textit{in } S(\Gamma(\Delta)) \textit{ such that } s \neq s'. \\ U_+(\Delta) - V(\alpha',\Delta), & (\alpha \twoheadrightarrow \alpha') \in R(\Gamma(\Delta)), \textit{ and } (s,\alpha) \textit{ and } (s,\alpha') \textit{ both exists} \\ & \textit{in } S(\Gamma(\Delta)). \end{cases} \tag{5.3}$$

We use the notation $FP(\alpha,\Delta) = V(\alpha,\Delta) - T(\alpha,\Delta)$ for the examples that are the false positives of the argument α in the argumentation schema Δ, and $FN(\alpha,\Delta) = T(\alpha,\Delta) - V(\alpha,\Delta)$ for the examples that are the false negatives of the argument α in the argumentation schema Δ.

In the context of binary classifications, the coverage of an argument α can be seen as a classification that α does of its target in the argumentation schema's context. The target of an argument can then be seen as either the set of positive examples of its argumentation schema in the case of a root-argument, or the set of either false positives or false negatives for the classification done by an argument α' attacked by α otherwise. An agent A_k will agree upon the argument α if, to its knowledge, the coverage and the target of α are equal. The agreement function is formally defined below.

Definition 44 (Argument Agree Function)**.** *Let $\mathbb{G}$ be the set of all possible argumentation schemas. Then* agree: $Ag(\mathbb{G}) \times A(\Gamma(\mathbb{G})) \times \mathbb{G} \to \{\text{true, false}\}$ *is a function such that, for every argumentation schema $\Delta \in \mathbb{G}$, every agent $A_k \in Ag(\mathbb{G})$ and every argument α in $A(\Gamma(\Delta))$ yields a Boolean value:*

$$agree(A_k,\alpha,\Delta) = \begin{cases} \text{true}, & \textit{if } FP(\alpha,\Delta) \cap U_k = \emptyset \textit{ and } FN(\alpha,\Delta) \cap U_k = \emptyset. \\ \text{false}, & \textit{otherwise.} \end{cases} \tag{5.4}$$

In our argumentation model, we impose as an additional rule that an agent A_k can insert an argument α in an argumentation schema Δ only if $agree(A_k,\alpha,\Delta) = \text{true}$.

5.2.5 Generating Counter-Arguments

In an argumentation schema Δ between two agents A_1 and A_2, if an argument α such that (α belongs to A_k) $\in B(\Gamma(\Delta))$ subsumes a set $err \in \{FP(\alpha,\Delta), FN(\alpha,\Delta)\}$, then since α can only have been inserted in Δ if $agree(A_k,\alpha,\Delta) = \text{true}$, the intersection $err \cap U_k$ is always empty. According to Definition 36, we know that $U_O(\Delta)$ is the overall context of A_1 and A_2. According to Definition 43, we also know that $FP(\alpha,\Delta)$ and $FN(\alpha,\Delta)$ are subsets of $U_O(\Delta)$. This means that the true and false positives —and therefore the examples of *err*— are in the overall context of the agents. Moreover, according to Definition 24 that the overall context of our agents is the union of their two sets. Therefore, we know that if *err* is not empty, then $err \in U_{-k}$.

If an agent A_k does not agree upon an argument α, it can create an other argument α' to attack α in order to notify A_{-k} that the set $err \in \{FP(\alpha,\Delta), FN(\alpha,\Delta)\}$ is non-empty. We say that α' is a counter-argument of α. A counter-argument is an argument that targets either the false positives or negatives of the argument it attacks. We will now see how A_k can create a counter-argument α' such that $T(\alpha,\Delta) = err$. Once inserted in the argumentation schema, the argument α' will cause α to be marked as *rejected* by the marking function m defined in Definition 34. In order to create α', the agent will used either a set of examples or an intensional definition learned through the ABUI algorithm, introduced in Chapter 1 and presented below.

ABUI is an inductive learning algorithm that has two modes. Its intension generation mode is a mode that, for a set of generalizations AA called set of accepted arguments and two sets of positive and negative examples U^+ and U^-, will try to find an intensional definition I such that:

- $I \sqsubseteq U^+$ and $I \not\sqsubseteq U^-$, and
- For each each generalization $g \in I$ and each generalization $g' \in AA$, $g \not\sqsubseteq g'$.

The second mode of ABUI, called the argument generation mode, is a mode that for a generalization g, a set of generalizations AA called set of accepted arguments and two sets of positive and negative examples U^+ and U^-, will try to find a generalization g' such that:

- $g \sqsubseteq g'$, and
- $g' \sqsubseteq U^+$ and $g' \not\sqsubseteq U^-$, and
- For each generalization $g'' \in AA$, $g' \not\sqsubseteq g''$.

Definition 45 (ABUI Function). *Let $\mathbb{I}$ be the set of all possible sets of generalizations, $\mathbb{U}$ the set of all possible examples and $\mathbb{G}$ the set of all possible argumentation schemas. Then* ABUI: $A(\Gamma(\mathbb{G})) \times S(\Gamma(\mathbb{G})) \times \mathbb{U} \times \mathbb{G} \to \mathbb{I}$ *is a function such that for an argumentation schema Δ, an argument $\alpha \in A(\Gamma(\Delta))$, a set of examples U and a sign $s \in S(\Gamma(\Delta))$, $ABUI(\alpha, s, U, \Delta)$ yields:*

- $I = g_1 \cup \ldots \cup g_n$, *if* $I' = \{g'_1, \ldots, g'_n\}$ is intension of α *exists* $F(\Gamma(\Delta))$ *and if the argument generation mode of the* ABUI *algorithm can create a generalization g_i for each $g'_i \in I'$, such that:*
  - *$g_i \sqsubseteq g'_i$, and*
  - *for each $g'' \in \{I'' \in A'(\Gamma(\Delta)) | (I'', \alpha) \in F(\Gamma(\Delta)) \wedge (s', \alpha) \in L(\Gamma(\Delta))\}$, $g_i \not\sqsubseteq g''$, and*
  - *for each example $e \in U$, $g_i \sqsubseteq e$ and for each example $e' \in (U_O(\Delta) - \{e \in U | g_i \sqsubseteq e\})$, $g_i \not\sqsubseteq e'$.*
- *$I = \emptyset$, otherwise.*

The ABUI algorithm is useful to create an intensional definition that subsumes the false positives or false negatives made by an argument through inductive learning. With the ABUI algorithm, an agent can create a counter argument for any argument by using the *attack* function defined below.

Definition 46 (Attack Generation). *Let $\mathbb{I}$ be the set of all possible sets of generalizations, $\mathbb{U}$ the set of all possible sets of examples, $\mathbb{A}$ the set of all possible sets of arguments and $\mathbb{G}$ the set of all possible argumentation schemas. Then* attack: $Ag(\mathbb{G}) \times A(\Gamma(\mathbb{G})) \times \mathbb{G} \to \mathbb{A}$ *is a function such that, for an argumentation schema Δ, an agent $A_k \in Ag(\Delta)$, and an argument α in $A(\Gamma(\Delta))$, $attack(A_k, \alpha, \Delta)$ yields a set of arguments $(P \cup N)$, where P and N are defined as follows:*

1. *P is the a of arguments such that:*
   - *$P = \emptyset$ if $FP(\alpha, \Gamma) = \emptyset$, and*
   - *otherwise:*
     - $P = \{\langle \alpha, s', A_k, I \rangle\}$ *if* ABUI*$(\alpha, s', FP(\alpha, \Delta) \cap U_k, \Delta)$ yields a non-empty intensional definition I, and*
     - $P = \{\langle \alpha, s', A_k, FP(\alpha, \Delta) \cap U_k \rangle\}$ *otherwise.*
2. *N is a set of arguments such that:*
   - *$N = \emptyset$ if $FN(\alpha, \Gamma) = \emptyset$, and*
   - *otherwise:*
     - $N = \{\langle \alpha, s, A_k, I \rangle\}$ *if* ABUI*$(\alpha, s, FN(\alpha, \Delta) \cap U_k, \Delta)$ yields a non-empty intensional definition I, and*
     - $N = \{\langle \alpha, s, A_k, FN(\alpha, \Delta) \cap U_k \rangle\}$ *otherwise.*

5.2.6 Argumentation Schema Integration

Argumentation for Concept Creation Setup

An argumentation in the context of concept creation takes place during one step of the global argumentation over the meaning that our agents have in our model. Once having decided how to determine if an example e should belong to the adjunct set U^+ of the new concept's intensional definition I or not as explained in the beginning of this section, two agents A_1 and A_2 can create a new argumentation schema $\Delta_0 = \langle A_1, A_2, \Gamma_0, U^+ \rangle$ with an argumentation tree $\Gamma_o = \langle Ag = \{A_1, A_2\}, S = \{+, -\}, A_0, A'_0, R_0, B_0, L_0, E_0, F_0 \rangle$, such that all sets in Γ_0 are empty. We will now explain how the agents can, turn by turn, add new arguments in the argumentation schema Δ_0 that will change both Δ_0 and its argumentation tree, until the agents find after n turns an argumentation schema $\Delta_n = \langle A_1, A_2, \Gamma_n, U^+ \rangle$ such that the root-argument α_r of Γ_n represents an acceptable binary classification for U^+, such that I is intension of α_r in Δ. At the beginning of the argumentation in the context of concept creation, the agents should have decided which one of them will lead the argumentation, according to the protocol described in Section 5.5.5. We will consider A_1 to be the lead for the rest of the section. A_1 will be in charge of the creation of the root-argument α_r, while A_2 will support A_1 during the argumentation.

The general idea remains the same as for the rest of the general argumentation. The agent with the token inserts or deletes arguments in the argumentation schema, send messages and passes the token. The agents exchange their arguments through messages. Each agent A_k keeps an instance of their current argumentation schema Δ_t in memory, and sends a message *Insert-Argument*(α) or *Delete-Argument*(α) when it inserts or deletes an argument from Δ_t. This allows the agents to create Δ_{t+1} in parallel. When new examples are added to the argumentation schema, the agents add them to their contexts: for any argument α, if an agent A_k notices upon creating the argumentation schema Δ_{t+1} that there exists a set of examples U such that $(U, \alpha) \notin E(\Gamma(\Delta_t))$ and $(U, \alpha) \in E(\Gamma(\Delta_{t+1}))$, then A_k adds the examples from U to its context U_k.

First Turn of Argumentation for Concept Creation

During its first turn, the agent A_1 attempts to create a new root argument $\alpha_r = root(\Delta_0)$ using the function root defined below.

Definition 47 (Root Creation Function). *Let $\mathbb{G}$ be the set of all possible argumentation schemas, $\mathbb{A}$ the set of all possible root-arguments and* ABUI *the algorithm presented in Definition 45. The function* root: $Ag(\mathbb{G}) \times \mathbb{G} \to \mathbb{A}$ *is a function that, for each argumentation schema Δ and each agent A_k of Δ,* root(A_k, Δ) *yields:*

- $\langle +, A_k, I \rangle$*, if the intension generation mode of* ABUI *algorithm can create an intensional definition I such that for each generalization $g \in I$:*
  - *for each* $g' \in \{I' \in A'(\Gamma(\Delta)) | (I, \alpha) \in F(\Gamma(\Delta)) \wedge (-\text{ labels } \alpha) \in L(\Gamma(\Delta))\}$*,* $g \not\sqsubseteq g'$*, and*
  - *for each example* $e \in (U_+(\Delta) \cap U_k)$*,* $I \sqsubseteq e$ *and for each example* $e' \in ((U_O(\Delta) - U_+(\Delta)) \cap U_k)$*,* $I \not\sqsubseteq e'$*.*
- $\langle +, A_k, \emptyset \rangle$*, otherwise.*

If A_1 succeeds to create a root-argument α_r with a non-empty intensional definition, A_1 inserts α_r in the argumentation schema Δ_0 that becomes Δ_1. If A_1 does not succeed to create a root-argument with a non-empty intensional definition, the argumentation for concept creation stops. During its first turn the agent A_2 does nothing.

Second Turn of Argumentation for Concept Creation

During its second turn, the agent A_2 verifies whether or not it agrees with the root α_r of Δ_1 by checking the value of $agree(A_2, \alpha_r, \Delta_1)$. If $agree(A_2, \alpha_r, \Delta_1)$ = true, the argumentation for concept creation stops on a success as the intension of α_r, being agreed upon by both agents, is an intensional definition that classifies the examples U^+ without any false positives or false negatives in the overall context U_O. If $agree(A_2, \alpha_r, \Delta_1)$ = false, A_2 inserts the counter arguments from $attack(A_2, \alpha_r, \Delta_1)$ in the argumentation schema Δ_1.

T$^{\text{th}}$ Turn of Argumentation for Concept Creation

If A_2 agrees upon the root $\alpha_{r,t}$ of Δ_t during its t^{th} turn, then the argumentation for concept creation stops on a success. Each agent agent A_k starts its t^{th} turn by creating two sets of arguments: *agreed* and *disagreed*. Then, A_k searches for the *leaves* of Δ_t, defined in Definition 48.

Definition 48 (Leaves)**.** *For any argumentation schema Δ, the leaves of Δ are the arguments from the set:*

$$leaves(\Delta) = \{\alpha \in \Gamma(\Delta) | (\alpha' \twoheadrightarrow \alpha) \notin R(\Gamma(\Delta))\}.$$

For each argument $\alpha \in leaves(\Delta_t)$, the agent A_k creates a set of arguments $obsolete_\alpha$. Then, if $agree(A_k, \alpha, \Delta_t)$ yields true, the agents A_k adds the argument α to the set *agreed* since both agents have agreed upon α, and every argument α'' such that $(\alpha \twoheadrightarrow \alpha'') \in R(\Gamma(\Delta))$ to the set *obsolete* since both agents have agreed upon an argument that defeats α''. On the contrary, if $agree(A_k, \alpha, \Delta_t)$ yields false, the agent A_k adds the argument α to the set *disagreed* since α classifies its target with either false negatives or positives. Once all the leaves of Δ_t have been either agreed or disagreed upon, the agent A_k deletes the arguments of each set $obsolete_\alpha$ from the argumentation schema Δ_t, which becomes the argumentation schema Δ_t^1. Then, A_k repeats the same process with the leaves of Δ_t^1, and continues until it reaches an argumentation schema $\Delta_t^n = \Delta_t^{n-1}$.

Once the argumentation schema Δ_t^n has been reached, the agent A_k adds all the arguments from *agreed* to the set of accepted arguments $A'(\Gamma(\Delta_t^n))$ since they have all been agreed upon by both agents. Then, for each argument α in *disagreed*, A_k inserts the counter arguments from $attack(A_k, \alpha, \Delta_t^n)$ in the argumentation schema Δ_t^n, which becomes the argumentation schema Δ_{t+1}. If the agent A_1 notices that Δ_{t+1} has no root during its t^{th} turn, A_1 attempts to create a new root-argument $root(\Delta_{t+1})$ and to insert it in the argumentation schema. If A_1 does not succeed to create a root with a non-empty intensional definition, the argumentation schema for concept creation stops.

5.3 General Structure of Argumentation

5.3.1 Argumentation Goal

When two agents meet, they are prepared to face a situation where they do not understand each other. In order to be ready for argumentation, they both create a copy of their initial contrast set. This copy of their initial contrast set becomes their *current* contrast set. These current contrast sets might be modified later if the agents start an argumentation. The goal of an argumentation is always to make two agents reach mutual intelligibility without changing their contrast sets in a non-monotonic way. While the agents can chose different strategies to reach the mutual intelligibility with this constraint, all strategies have the same final goal and also share similar intermediary goals.

While Section 4.5 presents the characteristic of mutual intelligibility, this section will focus on its achievement from a point where our system of two agents is unable to guarantee it. In this section, we stated that while the synchronic agreement was probably not initially reached by the agents, the diachronic agreement is initially always found. The diachronic agreement being always initially found is due to the fact that this agreement symbolize the similarity between the initial and the current contrast set. Since the current contrast set is initially a copy of the initial contrast set, there is initially no difference between the two contrast sets and therefore no diachronic disagreement.

5.3.2 Argumentation Turns

Our argumentation model is a turn-by-turn model, meaning that only one agent can take take actions at a given time. In order to synchronize the turns, the agents have a *token*. When an agent gets the token, it takes as many actions as it needs to, and then passes the token to the other agent which does the same. The beginning of an argumentation on meaning always starts with the experimenter giving the token to a random agent. The duration which goes from an agent receiving the token and passing it is called the *turn* of this agent.

A turn is always organized following the same structure: first the agent receives the token, then the agent reads its messages, then the agent updates its knowledge according to the new elements received in the messages, then the agent take the actions that are dictated by its current state and its current knowledge, then the agent eventually sends messages to the other agent, then the agent sets its next step, and finally the agent passes the token to the other agent.

5.3.3 Argumentation Steps

An argumentation between two agents is always cyclic. At each iteration of the cycle, the agents are closer from the mutual intelligibility than they were during the last iteration – even if sometime they have more synchronic disagreements than during the last iteration. Each cycle of the argumentation can be divided in steps. In each step, the agents pass by a number of states that determine the agents' actions. Once an agent has taken all the actions that are determined by its current state, it might changes its state and let the other agent take actions. In order to keep track of which agent should take action, the agents have one token. An agent can act only if it has the token. The last action of an agent before an action of the other agent is always passing the token, and if an agent should change its state it always does so as the last action before passing the token.

Each argumentation strategy can be split into four main steps. The first step is to compute some overall pairing relations between concept(s) from different agents. The second step is to infer disagreements from the pairing relations computed during the first step, and to list them. The third step is to pick one disagreement to resolve, and the fourth step is to resolve the disagreement picked during the third step. Once the fourth step is over, the argumentation goes back to the first step in a new iteration of its cycle.

5.4 Argumentation Strategies

The different strategies on argumentation over the meaning diverge by their approach to disagreement identification. The first strategy, called the systematic strategy requires from the agents to exchange all of their intensional definitions when they meet each others. This ensures that two agents using the systematic strategy start their argumentation with knowledge over all the synchronic disagreements between the two agents' initial contrast sets.

The second approach is called the lazy approach. In this approach, the agents are starting their communication with a naming game: an example is presented to them and the agents name this

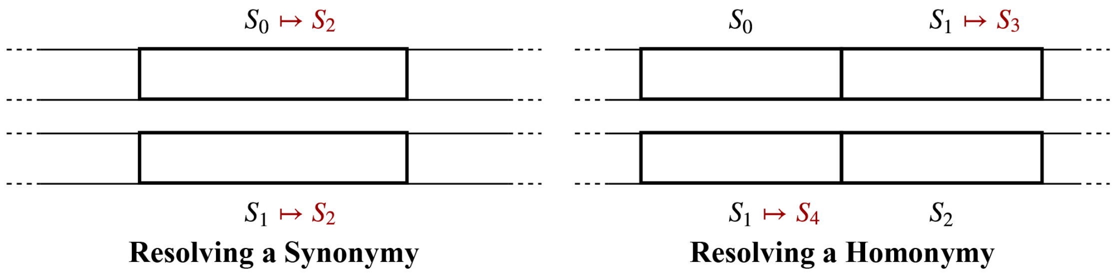


Figure 5.5: Resolution of synonymy and homonymy disagreements by sign replacement.

examples. By comparing the sets of signs used by the agent to name the example, the two agents can infer if there is a synchronic disagreement between them.

5.5 Resolution of Disagreements

A disagreement can involve a maximum of two concepts. A disagreement might be partially caused by the signs of these two concepts, as it is the case for the lexical disagreements, however all types of disagreements are based on the pairing relation between the two concepts. Since removing one of the concepts from its contrast set removes the pairing relation between them at the same time, removing a concept involved in a disagreement resolves the disagreement.

However, the examples that were covered by a concept that has been removed would not be covered anymore. For this reason, a concept that is removed in order to resolve a disagreement should be replaced by a set of concepts that are not causing synchronic or diachronic disagreements. These new concepts should, as much as they can, cover the examples that were covered by the concepts they are replacing. In this section, we present how our model replaces concepts in disagreements by concepts that are not.

5.5.1 Resolving Lexical Disagreements

In the case of a lexical disagreement, the partition made by the two concepts involved in the disagreement are not at fault. The two concepts have a pairing relation of equivalence (homonymy) or are disjoint (synonymy), and the only thing leading them to cause a disagreement is their signs. Therefore, a concept involved in a lexical disagreement is replaced by a concept that share the same intensional and extensional definition, but that has a different sign.

Resolving Synonymy Disagreements

Let C_1 and C_2 be two concepts from two agents A_1 and A_2, such that their pairing relation in the overall context U_O is $C_i \equiv_{UO} C_j$ and their signs are different: $s(C_i) \neq s(C_j)$. According to Section 4.5.3, C_i and C_j are causing a synonymy disagreement d_s. In order to resolve d_s, each agent A_k replaces its concept $C_k = \langle s_k, I_k, E_k \rangle$ by a new concept $C'_k = \langle s, I_k, E_k \rangle$. This process is represented in Figure 5.5 (left), where two concepts having different signs s_0 and s_1 are replaced by two concepts having the same sign s_2. The resulting concepts C'_1 and C'_2 are still in a relation of equivalence, since their intensional and extensional definitions remained the same as C_1 and C_2, and their signs are now the same. Therefore, according to Section 4.5.2, the pair of concepts C'_1, C'_2 is not causing a disagreement. As mentioned in Section, 5.1.3, the new sign is different from the signs of the overall vocabulary. Therefore, the new concepts cannot cause a homonymy disagreement.

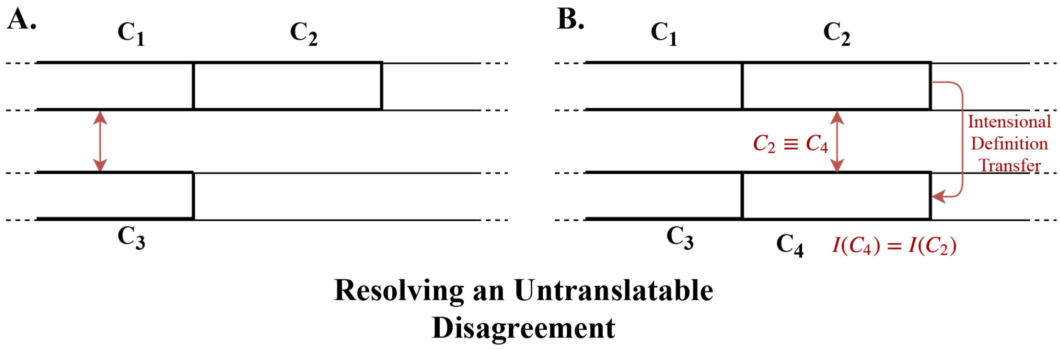


Figure 5.6: Resolution of an untranslatable disagreement by creating a copy C_4 of the concept C_2 in the bottom contrast set, where C_2 had previously no equivalent.

Resolving Homonymy Disagreements

Let C_1 and C_2 be two concepts from two agents A_1 and A_2, such that their pairing relation in the overall context U_O is $C_i \oslash_{UO} C_j$ and they share the same sign: $s(C_i) = s(C_j)$. According to Section 4.5.3, C_i and C_j are causing a homonymy disagreement d_s. In order to resolve d_h, each agent A_k replaces its concept $C_k = \langle s, I_k, E_k \rangle$ by a new concept $C'_k = \langle s_k, I_k, E_k \rangle$. This process is represented in Figure 5.5 (right), where two concepts having the same sign $s_1$are replaced by two concepts having different signs s_3 and s_4. The resulting concepts C'_1 and C'_2 are still in a relation of disjunction, since their intensional and extensional definitions remained the same as C_1 and C_2, and their signs are now different. Therefore, according to Section 4.5.2, the pair of concepts C'_1, C'_2 is not causing a disagreement. As mentioned in Section, 5.1.3, the two new signs are different from the signs of the overall vocabulary. Therefore, the new concepts cannot cause a synonymy disagreement.

5.5.2 Resolving Untranslatable Disagreements

The untranslatable disagreements are a special scenario, as they are not involving two concepts, but one concept and the absence of its equivalent in the other contrast-set. Therefore, an untranslatable disagreement is not resolved by removing a concept but by creating a new one, equivalent to the concept that has no equivalent. Let $C = \langle s, I, E \rangle$ a concept of the agent A_1, such that the agent A_2 has no concept C' in its contrast set such that $C \equiv_{UO} C'$. According to Section 4.5.3, this situation results in an untranslatable disagreement d_u. In order to resolve d_u, the agent A_2 creates a new concept $C' = \langle s, I, Adj(I, U_2) \rangle$. Since the two concepts C and C' share their intensional definition – and thus their adjunct set, they are equivalents according to Definition 22. Now that there exists a concept C' in the contrast-set of A_2 such that $C \equiv_{UO} C'$, the situation does not cause an untranslatable disagreement anymore. Since the sign of the concepts C and C' are the same, the two equivalent concepts are not causing a synonymy disagreement.

5.5.3 Resolving Self-Disagreements

Let C_1 and C_2 two concepts from an agent A_1 such that $C_1 \otimes_{UO} C_2$. It is important here to note a few things. First of all, since C_1 and C_2 belong to a same contrast set, the agent A_1 cannot see their local pairing relation as $C_1 \otimes_{U1} C_2$, but only as $C_1 \oslash_{U1} C_2$. This means that A_1 has interacted with another agent A_2, such that A_2 has in its local context U_2 some examples subsumed by both $I(C_1)$ and $I(C_2)$. Moreover, the only pairing relation that can be involved in a self-disagreement is

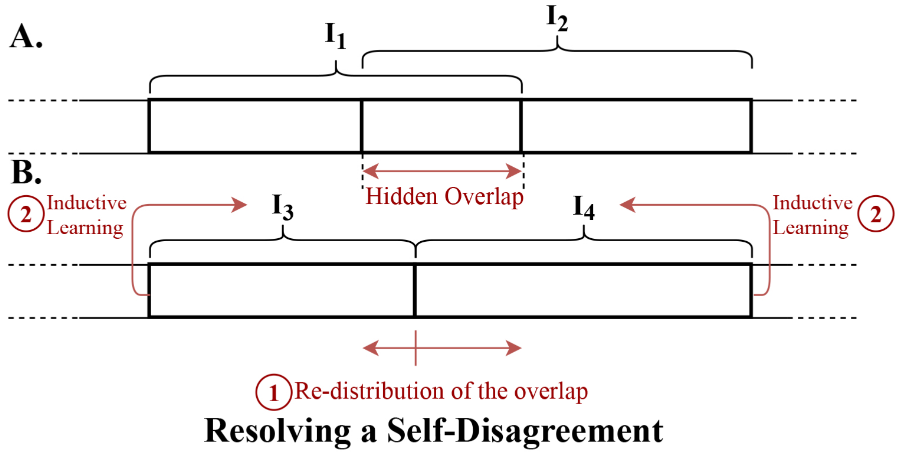


Figure 5.7: Resolution of a self-disagreement in two steps. First, the examples from the overlap of the two concepts are re-distributes according to their anti-unification distance to each concept's intensional definition, and then inductive learning is used to create a new pair of concepts that respects the new example distribution.

an overlap. Indeed, if A_1 sees the relation between its two concepts as $C_1 \oslash_{U1} C_2$, this means that there are examples both subsumed by $I(C_1)$ and not $I(C_2)$, and examples subsumed by $I(C_2)$ and not $I(C_1)$ – in the local and the overall context. Therefore, the overall pairing relation between C_1 and C_2 is either an overlap or a disjunction. Since the signs of the concepts from a same contrast set are all different, the only possible pairing relation causing a disagreement is the overlap. Therefore, a self-disagreement always involves two overlapping concepts.

An important specificity of the self-disagreement, is that the examples that are in the overlap of the two concepts have no particular reason to belong to one concept or another. Indeed, the agent A_1 does not know about these examples, therefore their classification cannot affect its synchronic or diachronic agreement. Moreover, while the agent A_2 might classify these examples in two or more different concepts, the resulting disagreements should be handled as separate semantic disagreements, not as self-disagreements. For this reason, the two concepts C_1 and C_2 should be replaced by a new pair of concepts C_1', C_2' such that $C_1' \odot_{UO} C_2'$ and $Adj(C_1', U_O) \cup Adj(C_2', U_O) = Adj(C_1, U_O) \cup Adj(C_2, U_O)$.

The two new concepts will be created through argumentation, one after the other. But the first step for the two agents, is to decide of which sign should be associated to each of the elements from the intersection of $Adj(C_1, U_O)$ and $Adj(C_2, U_O)$. For now, these examples are associated to both $s(C_1)$ and $s(C_2)$, which prevents the creation of a new concept through argumentation. Each agent starts by creating the sets of examples:

- $U_{k,C1,\overline{C2}} = Adj(C_1, U_k) - Adj(C_2, U_k)$
- $U_{k,C1,C2} = Adj(C_1, U_k) \cap Adj(C_2, U_k)$
- $U_{k,\overline{C1},C2} = Adj(C_2, U_k) - Adj(C_1, U_k)$

In order to choose, for each example from $U_{1,2,k}$, which sign of $s(C_1)$ or $s(C_2)$ is better suited, each agent uses the anti-unification similarity measure, written AU similarity measure. The AU similarity measure is based on the anti-unification distance measure (Ontañón and Plaza, 2012), which measures the number of steps needed to find the anti-unification of two feature terms. in our model, the AU measure is used to quantify the similarity between an intensional definition g and an example e, written $dAU(g, e)$. Let $I_1 = \{g_{1,1}, \ldots, g_{1,m}\}$ the intensional definition of C_1 and

$I_2 = \{g_{2,1}, \ldots, g_{2,n}\}$ the intensional definition of C_2; For each example $e \in U_{1,2,k}$, A_k calculates the average similarities:

$$D_1 = \frac{1}{|I_1|} \times \sum_{i=1}^{m} dAU(g_{1,i}, e) \text{ and } D_2 = \frac{1}{|I_2|} \times \sum_{i=1}^{n} dAU(g_{2,i}, e)$$

Then, A_k creates a set of examples $E_{1,k} = \{e \in U_{1,2,k} | D_1 \geq D_2\}$ and a set of examples $E_{2,k} = \{e \in U_{1,2,k} | D_2 > D_1\}$. A_k then adds the examples from $U_{1,\overline{2},k}$ to $E_{1,k}$ and the examples from $U_{\overline{1},2,k}$ to $E_{2,k}$. Finally, A_k associates all the examples of $E_{1,k}$ with $s(C_1)$ and all the examples of $E_{2,k}$ with $s(C_2)$. Since the examples of $U_{1,2,k}$ have been re-distributed, the set of associations $E_{1,k} \mapsto s(C_1) \cup E_{2,k} \mapsto s(C_2)$ is coherent. Moreover, since the agents have both access to the intentional definitions I_1 and I_2, an example e that is present in both $U_{1,2,1}$ and $U_{1,2,2}$ will have the same associated distances D_1 and D_2 independently of the agent that measures them. For this reason, if the example e is put in the set $E_{1,x}$ by A_1, it will be put in the set $E_{2,x}$ by A_2 and therefore associated to a same sign. For this reason, the set of associations $E_{1,1} \mapsto s(C_1) \cup E_{1,2} \mapsto s(C_1) \cup E_{2,1} \mapsto s(C_2) \cup E_{2,2} \mapsto s(C_2)$ is also coherent, authorizing the agents to learn new concepts through argumentation.

The new concepts C_1' and C_2' are created through argumentation, sequentially. The agents will start by creating the new concept C_1' that will replace C_1. Following the protocol described in Section 5.1.3 will lead to the agent A_1 supervising the creation of C_1'. A_1 will therefore create a new belief α_1 using the set of examples $E_{1,1}$ as a set of positive examples. A_2 will evaluate this belief using $E_{2,1}$ as the set of positive examples, and eventually argue with A_1 until it eventually accepts a belief $\alpha' = \langle +, I_1', A_1 \rangle$. Once the belief is accepted, A_1 creates the new concept $C_1^{1'} = \langle s(C_1), I_1', E_{1,1} \rangle$ and replaces C_1^1 with it in its contrast set. Similarly, A_2 creates a new concept $C_1^{2'} = \langle s(C_1), I_1', E_{2,1} \rangle$ and replace C_1^2 with it in its hypothesis. Adopting the same strategy for the creation of C_2', the agent A_1 now has a pair of concepts C_1', C_2' that are not overlapping in its contrast set.

Of course, the re-distribution of the examples that are belonging to both of the two adjunct sets $Adj(C_1, U_O)$ and $Adj(C_2, U_O)$ could be different. Each example e could be randomly assigned to one of the two new concepts C_1' and C_2', however this would force the agent A_k associating e with the sign $s(C_k)$ to send the association $e \mapsto s(C_k)$ to the other agent, thus increasing the number of examples exchanged. Without exchanging e, the two agents would risk that A_{-k} also has the example e in its contrast-set, and associates it with another sign $s(C_{-k})$. This would result into an non-consistent set of associations on which to build the new intensional definitions upon, which is not possible according to Section 5.1.3.

5.5.4 Resolving Semantic Disagreements

Semiotic disagreements are resolved through the refinement of pre-existing concepts, which means that the concepts involved in a semantic disagreements will either be removed from the contrast set or replaced by a set of co-hyponyms that are partitioning the examples of the concepts involved in the disagreement.

Resolving Indistinguishable disagreements

The indistinguishable disagreement is a particular type of example that can only appear if the model admits an error threshold τ_E. The notion of error threshold is presented later in Chapter 6. An error threshold changes the pairing function of our model to neglect non-empty partial sets of comparatively small sizes during the computation of r-triplets. If two newly created concepts C_1 and C_2 both have adjunct sets such that $Adj(C_1, U_O) \geq \tau_E$ and $Adj(C_2), U_O \geq \tau_E$ but:

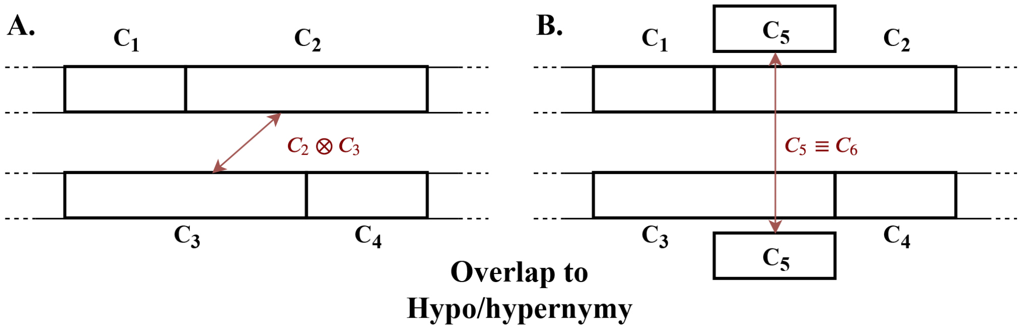


Figure 5.8: Resolution of an overlap into two hypo/hypernyms. The overlap between the two concepts is used to create the extensional definition of a new concept, hyponym of both overlapping concepts.

- $U_{O,C_1,C_2} < \tau_E$ and
- $U_{O,\overline{C_1},C_2} < \tau_E$ and
- $U_{O,C_1,\overline{C_2}} < \tau_E$

then, the two concepts are considered to be too close from equivalence and one should be removed. The agents resolve the indistinguishable disagreement by removing the concept $C \in \{C_1, C_2\}$ with the smallest adjunct set $Adj(C, U_O)$ from the concerned contrast sets. Since C is removed from the argumentation, the disagreement caused by the relation between C_1 and C_2 is resolved. This resolution causes a loss in the example coverage of the final contrast set.

Resolving Overlap disagreements

Let C_2 and C_3 be two concepts from two agents A_1 and A_2, such that their pairing relation in the overall context U_O is $C_2 \otimes_{UO} C_3$. According to Section 4.5.3, C_2 and C_3 are causing an overlap disagreement d_o. The transformation of the overlap disagreement d_o into two hypo/hypernymy disagreements is represented in Figure 5.8, and the further resolution of all three semantic disagreements is represented in Figure 5.9.

In order to resolve d_o, A_1 and A_2 will create a concept C_5 that is the hyponym of both C_2 and C_3. The adjunct set of the concept C_5 will be the set of examples from the overall context that are subsumed by $I(C_2)$ *and* $I(C_3)$. The agents use argumentation in order to create $I(C_5)$, as described in Section 5.1.3. Since neither C_2 nor C_3 is removed from the agents contrast sets, the disagreement d_o is *not* resolved. Moreover, at least two new disagreements appeared. Since the adjunct set of C_5 is a subset of the adjunct sets of C_2 and C_3, we have:

- $C_1 \odot_{UO} C_5$, and
- $C_2 \odot_{UO} C_5$.

According to Section 4.5.3, this situation causes two hypo/hypernymy disagreement d_{h2} where C_2 is the hypernym, and d_{h3} where C_3 is the hypernym. However, it is explained in Section 5.5.4 below that resolving a hypo/hypernymy disagreement involves the removal of the hypernym. Therefore, resolving either d_{h2} or d_3 will also resolve the disagreement d_o.

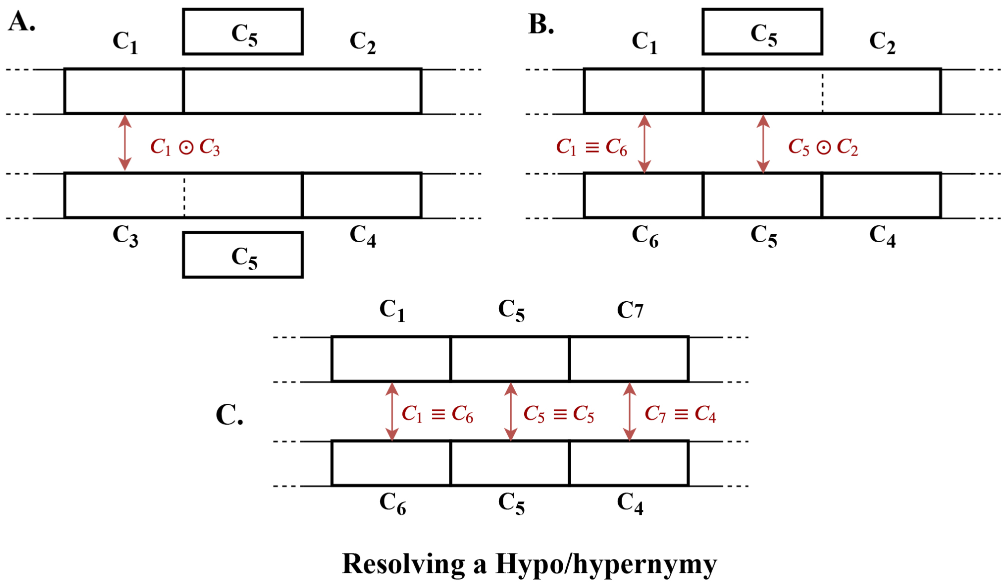


Figure 5.9: Resolution of two hypo/hypernymies resulting from the resolution of the overlap presented in Figure 5.8. The agents are creating the co-hyponyms of the concept created to resolve the overlap. The removal of the hypernyms resolves both the two hypo/hypernymies disagreements and the overlap disagreement.

Resolving Hypo/Hypernymy disagreements

Let $C_1 \in S_1$ and $C_3 \in S_2$ be two concepts from two agents A_1 and A_2, such that their pairing relation in the overall context U_O is $C_1 \odot_{UO} C_3$, and let C_3 be the hypernym of C_1. According to Section 4.5.3, C_1 and C_2 are causing a hypo/hypernymy disagreement d_h. The resolution of the hypo/hypernym d_h is represented in Figure 5.9(A). In order to resolve d_h, A_2 replaces the hyponym C_3 with two co-hyponyms C_5 and C_6 in its contrast set. The first co-hyponym, C_6, is a copy of the hyponym C_1 where the sign $s(C_1)$ has been substituted for a new sign s_6 such that:

$$C_6 = \langle s_6, I(C_1), Adj(C_1, U_2)\rangle.$$

The second co-hyponym, C_5, is created through argumentation in order to that its adjunct set contains all the examples of $Adj(C_3, U_O)$ that are not in the adjunct set $Adj(C_6, U_O)$. Once the two co-hyponyms are created, they come to replace C_3 in the contrast set K_2. The removal of C_3 from K_2 resolves immediately the disagreement d_h.

5.5.5 Choosing Lead of Argumentation

In cases where the agents A_1 and A_2 needs to create a new concept through argumentation in order to resolve a disagreement, one of the two agents will take the lead of the argumentation in the context of concept creation. When an agent A_k is the first to enter a state where in has to decide whether of not it should take the lead, it takes the lead the agent A_{-k} has not already taken the lead, and if:

- the disagreement to resolve is a self-disagreement of A_k, or if
- the disagreement is an untranslatable disagreement, where the untranslatable concept belongs to A_k's contrast set, or if:

– A_k is the only one that have examples that will belong to the concept to create, or (exclusive or), if the disagreement is a hypo/hypernymy,
– the hypernym belongs to A_k's contrast set.

This order insures that A_k is the best suited to create root-arguments during the argumentation in the context of concept creation.

5.5.6 Order of Resolution

Types of disagreements are resolved in order, as certain types of disagreements should only be resolved once there are no disagreements from other types in the argumentation anymore. Self-disagreements should always be resolved first, as they are preventing contrast sets to comply with Definition 14 by putting two concepts from a same contrast set in a relation that is not a relation of disjunction. Since our model has been designed to allow two agents to align two contrast sets, ensuring that the containers K_k and K_{-k} are indeed partitions and therefore valid contrast sets, is the first priority of our agents. If there are not self-disagreement to resolve, the agents should resolve indistinguishable disagreements in priority. Indistinguishable disagreements are disagreements that are resolved by directly removing one of the two concepts involved: if a concept C_i is involved in an indistinguishable disagreement d, it would not be optimal for the agents to resolve a disagreement d' caused by two concepts C_i and C_j, eventually creating a new concept in the process, when d' is automatically resolved when C_i is removed during the resolution of d. If there are no indistinguishable disagreement to resolve, the agents should resolve the semantic disagreements. Since semantic disagreements involve the creation of new concepts and the removal of old ones, there is no point of resolving lexical or untranslatable disagreements involving the same concepts, when these concepts will be removed during the resolution of a semantic disagreement. Finally, untranslatable disagreements are resolved before lexical disagreements. Untranslatable disagreements still modify the set of concepts of the agents contrast sets, and for this reason, they need to be addressed before the agents change the sign of their concepts. Once the agents contrast sets make identical partitions of the overall context, the agents can address the lexical disagreements and change their signs.

5.6 Conclusion

This chapter uses the formalism defined in Chapter 4 to build our argumentation model. This model introduces our agents and their capabilities, and specifically how agents can create new concepts through a specific process. The structure of our argumentation is also presented and detailed, and the different strategies at the disposal of our agents to resolve encountered disagreements are listed and detailed.

The ideas presented in this chapter are valid for an error-free learning model. Later modifications that will in fact allow our argumentation model to assume an error degree in inductive learning. These modifications will be presented in Chapter 6.

Chapter 6

Inductive Learning Error Management

6.1 General Idea

Among the three semiotic elements of the concept, the intensional definition stands aside. Unlike the two other semiotic elements — the sign and the extensional definition — the intensional definition is not initially present in the data used by the agents to create concepts: it is the element that has to be learned. The intensional definition is created by the agent through inductive learning, which means that it can suffer from some of the limitations that are frequently encountered in symbolic machine learning. Each intensional definition is an attempt at a binary classification (see Section 5.1.3), and this classification suffers from two limitations associated with every machine learning classifiers:

1. it requires a certain number of positive and negative examples to be able to learn, and
2. it is likely to produce false positives and negatives (some degree of error is unavoidable)

More generally, we discussed in Section 5.1.3 that when an agent creates a concept C from right-path associations, it needs a class $U(\mapsto s)$ and its associated sign s. In order to be able to use inductive learning, the number of examples in the class $U(\mapsto s)$ should be above a certain threshold. We will call this threshold τ_1. Upon learning the new concept C from the right-path associations $U \mapsto s$, the set of examples $U(\mapsto s)$ should become the extensional definition of C, admitting that the classification ended with an accuracy of 1. Therefore, since $U(\mapsto s) \geq \tau_1$, the extensional definition $E(C)$ is expected to have at least τ_1 examples. Therefore, during their argumentation, the agents should not consider possible to have concepts that are expected to have less than τ_1 examples —and moreover they should never try to create a such concept. This issue is at odds with the fact that, until now, agents can consider pairing partial sets with a minimum of one example as non empty —which can cause them to try to create concepts for these partial sets. This can for instance be the case during the resolution of a semantic disagreement, as we discussed in Section 5.5.

The situation described in the paragraph above assumes that the inductive learning can be achieved with a perfect accuracy (a zero error degree), which is, as we mentioned earlier, unlikely to happen – especially because one agent is expected to have access to only roughly half of the examples of the overall contrast set during the right-path association learning of one of its initial concepts, which cannot guarantee a good accuracy over the other half of the overall context.

Let A_1 and A_2 be two agents. Each agent receives an equal and homogeneous partition of a set of left-path associations $U \mapsto S = \{s \dots s_n\}$. Each agent A_k tries to learn a new concept $C_k = \langle s, I_k, E_k \rangle$. The intensional definition I_1 learned by the agent A_1 is likely to either not subsume examples of A_2's context that A_2 associates with s (first-type error), or to subsume examples from A_2's context that A_2 does not associate with s (second-type error). That's why upon receiving I_1 and building a copy of C_1 in its hypothesis, A_2 cannot assume that the examples of $U_{2,C1,\overline{C2}}$ and $U_{2,\overline{C1},C2}$ are examples from a non-equivalent concept of A_1 as it should be the case with the pairing function presented in Section 4.4.1. The agents need to decide of a threshold τ_2 for the first-type error, and a threshold τ_3 for the second-type error, under which the agents assume that the size of a pairing partial set is not indicative of some hidden concept, but of a degree of error in the accuracy of the inductive learning method.

Instead of having three different thresholds, we chose to consolidate them in a single error threshold, since they all aim to change the same thing: the minimal amount of examples in a pairing partial set for it to not be considered as empty. This single threshold τ is the highest from the minimal concept size expected τ_1, and the maximal error tolerated from our classifications $\tau_2 + \tau_3$. In our scenarios and experiments, this threshold τ is given as a parameter to both agents before the communication starts. While it is probably possible to grant the multi-agent system the ability to find the optimal value for τ by itself, this is not part of our current research goals and we will not address it. In the general case where concepts from a same contrast set have roughly the same number of examples, $\tau_2 + \tau_3$ is expected to be inferior to τ_1 —the size of a concept should be greater than the typical classification error, otherwise the classifier's performances are so poor that it should not be use in the first place— and therefore we can consider that our threshold can be $\tau = \tau_1$ in most practical cases.

There are two aspects to be noticed concerning the error threshold. The first is that the error threshold cannot be less than one, as an error threshold of zero or less would imply that pairing partial sets considered as empty have a negative cardinality, which is strictly impossible in our model. The second is that with an error threshold $\tau = 1$, our model remains as it was before the introduction of the error threshold. An empty set is a set with no examples, an any number of example above one results in the set being considered as non-empty.

At this point, it is important to emphasis the fact that we consider the pairing partial sets to be empty if their cardinality is below the threshold τ purely from the point of view of our pairing function. The empty set $\emptyset$ still remains a set with a 0 cardinality.

However, for two concepts C_1 and C_2 such that:

- $U_{O,C1,C2} < \tau$
- $U_{O,\overline{C1},C2} < \tau$
- $U_{O,C1,\overline{C2}} \geq \tau$

they will now be considered as equivalent, $C_1 \equiv_{UO} C_2$, as if $U_{O,\overline{C1},C2}$ and $U_{O,C1,\overline{C2}}$ were empty while using our initial error-free pairing function.

The introduction of the error threshold, and the transformation that it induces in the notion of equivalence between concepts, requires the modifications of several definitions. An implementation of our model with the definitions below and an error threshold $\tau > 1$ is referred as an *error-tolerant model*, where the r-triplets will now store the cardinality of the pairing partial sets instead of "0" for empty and "1" for non-empty. On the other hand, an implementation of our model with the definitions presented in Chapter 4 is referred as a *error-free model*, where the r-triplets store "1" and "0" as Boolean variables for the truth value of the proposition: *my corresponding pairing partial set is non-empty.*

6.2 Effect on Semiotic Elements and Containers

Definition 11 stated that, for any concept C_i, we should have the equality $E(C_i) = X(C_i)$ where $X(C_i) = \{e \in U | I(C_i) \sqsubseteq e\}$. If the concept C_i has been created from left-path associations, this definition is not problematic since:

$$E(C_i) \Leftrightarrow Adj(C_i, U) \Leftrightarrow \{e \in U | I(C_i) \sqsubseteq e\} \Leftrightarrow X(C_i)$$

The equality $E(C_i) = X(C_i)$ is maintained and our definition still stands. However, if the concept is created from right-path associations or through argumentation, the situation is different. In each of these two situations, the extensional definition $E(C_i)$ of the created concept is the subset $U_+ \subset U$ that served as positive examples during inductive learning. As we now admit a certain degree of error during our inductive learning, the introduction of τ should be reflected in the relation between $E(C_i)$ and U_+ that can no longer be considered an equality relation.

A solution would be to modify the equality $E(C_i) = X(C_i)$ in the definition of concepts, but we prefer to actually modify the creation process of our concepts, in order to keep the equality $E(C_i) = X(C_i)$ and prioritize the left-path associations over the right-path associations in our model. If the model admits an error threshold $\tau > 1$, any concept created in a context U should have its extensional definition $E(C_i)$ to be $E(C_i) = Adj(C_i, U)$.

However, the decision to keep $E(C_i) = X(C_i)$ impacts directly the contrast sets, that cannot be longer strict partitions. Given the consistent set of right-path associations $U \mapsto S$, such that $S = \{s_1, \ldots, s_n\}$; if an agent A learns a concept C_i for each class $U(\mapsto s_i)$ with $s_i \in S$, then we can now tolerate that 1) $I(C_i)$ subsumes up to τ_E examples that are in the set $U - (U(\mapsto s_j))$ and 2) $I(C_i)$ does not subsume τ examples that are in $U(\mapsto s_i)$. While the fact that $U \mapsto S$ is consistent means that $(U, \{C(s_i) | s_i \in S\})$ should be a contrast set if the inductive learning takes place with a perfect accuracy during the creation of each C_i, this is not the case if the learning admits a degree of error τ_E. Actually, if both C_i and C_j have bean created with a first-type error of τ, the overlap $E_i \cap E_j$ can reach $2 \times \tau$, but we cap the maximum size of the intersection to τ for the sake of simplicity. The container $(U, \{C(s_i) | s_i \in S\})$ is not a contrast set anymore, as it is not a proper partition. We must change our definition of a contrast set such that:

- regarding the first point, the intersection between two extensional definitions of two different concepts can be non-empty as long as it is not above the threshold τ, and
- regarding the second point, we cannot longer guarantee that $E_1 \cup \ldots \cup E_n = U$.

Taking these two limitations into account, we can now give a new definition to contrast sets, Definition 49 below. This definition applies, instead of Definition 14, in implementations that use an error-tolerant model.

Definition 49 (Contrast Set With Degree of Error)**.** *A contrast set* $K = (U_K, S_K = \{C_1, \ldots, C_n\})$ *is a container, such that: 1) for each pair of concepts* $C_i, C_j \in S_K$*, the property* $|E(C_i) \cap E(C_j)| < \tau$ *holds; and 2) the signs of the concepts must be all pairwise different:* $\forall C_i, C_j \in K$*,* $i \neq j \Rightarrow s(C_i) \neq s(C_j)$*.*

Since no relation between sets of examples were required in the hypothesis, the modification to the definition of concept does not affect the definition of hypotheses. Therefore, the definition of a hypothesis is still Definition 15.

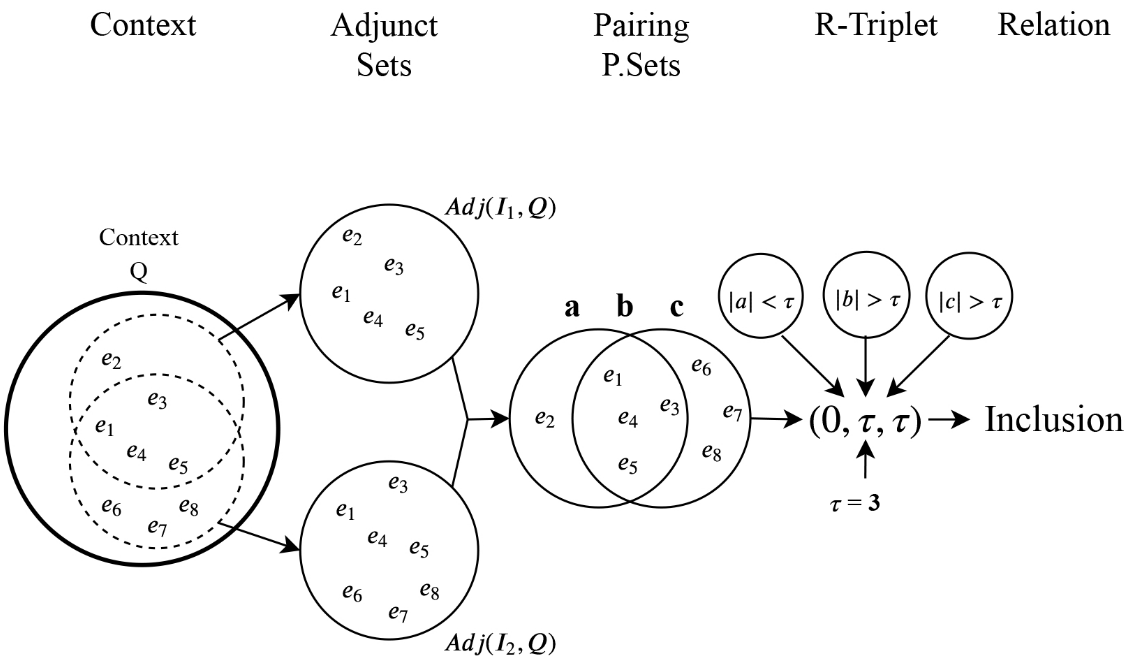


Figure 6.1: Computing the secured r-triplets of two concepts I_1 and I_2 in the context Q, while admitting a degree of error $\tau = 3$. This picture can be compared to Figure 5.1, which illustrated the same computation in an error-free.

6.3 Effect on Relations Between Concepts

The relations among concepts are extensively affected by the introduction of an error degree. We will move from the idea that τ as an element of a binary classification to an idea of error tolerance in order for two concepts to remain equivalents in a context. We move from a model where the equivalence between two concepts C_1 and C_2 in a context U is defined by $Adj(C_1, U) = Adj(C_2, U)$ and thus $|Adj(C_1, U) \triangle Adj(C_2, U)| = 0$, to a model where the equivalence between A and B is defined by $|Adj(C_1, U) \triangle Adj(C_2, U)| < \tau_E$.

6.3.1 Assuming a Degree of Error in R-Triplets

The first step to evaluate the relation between two concepts C_i and C_j in a context U was to find their pairing partial sets, as presented in Section 4.4.1. These pairing partial sets allowed us to find the r-triplet of C_i and C_j in U, but the definition of r-triplets was based on whether or not the partial sets were empty. Replacing the notion of set emptiness with the notion of set cardinal inferior to a threshold, we substitute Definition 4.4.1 by Definition 50 below:

Definition 50 (R-Triplet Function with Degree of Error). *Let ev be the function defined in Definition 20 and $g(U_x)$ be the function that, for every pairing partial set U_x, yields:*

- 1 *if* $|U_x| \geq \tau$, *and*
- 0 *otherwise.*

The function $r : \mathbb{X} \times \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{N}^3$, with $\mathbb{X}$ the domain of concepts and $\mathbb{U}$ the domain of all possible contexts, is a function that for each pair of concepts C_i,C_j and for a given context U yields a triplet $r(C_i, C_j, U) = (b_{-1}, b_0, b_1)$ called r-triplet, such that for $x \in \{-1, 0, 1\}$:

$$b_x = ev(x, C_i, C_j, U, g)$$

Using the r-triplet function defined in Definition 50 with the Definition 22 to find pairing relations between concepts results in more indulgent pairing relations. Small overlaps between concepts do not necessarily results into concepts not being equivalent. In this configuration, we say that our pairing relations are assuming a degree of error τ. However, Conjecture 2 does not hold after this substitution. Since Conjecture 2 was used to compute overall r-triplets from local r-triplets, the agents need to find another way to obtain their overall pairing r-triplet if they admit a degree of error τ in their argumentation.

6.3.2 Finding Overall R-Triplets in Error-Tolerant Models

In order to adapt our approach to incorporate degrees of error, we need to differentiate the aim of *local* r-triplets and *overall* r-triplets. While computing pairing partial sets and r-triplets, the agents' final goal is to find the overall r-triplet of a pair of concepts that will identify the overall pairing relation between these two concepts. The local r-triplets are just an intermediate step to help the agents finding the overall r-triplets without having to exchange their whole contexts.

When the model admits a degree of error, the agents will need more intermediate steps to determine their overall pairing relations. This will be reflected by new types of r-triplet that represent new steps of the transition between local pairing partial sets and overall pairing relations. The last type of r-triplet should be the same as the input of the pairing function presented in 4.4.1, that associates each possible Boolean triplet to a type of pairing relation.

Local R-Triplets

Let A_1 and A_2 be two agents, A_1 partitioning the context U_1 in the contrast set $K_1 = (U_1, S_1)$ and A_2 partitioning the context U_2 in the contrast set $K_2 = (U_2, S_2)$. Let $C_i \in S_1$ and $C_j \in S_2$ be two concepts. In order to find the pairing relation between two concepts C_i and C_j, the first step that the two agents can take is to find their local r-triplets. However, the local r-triplets now need to carry more information than what is defined in Definition 50. For this reason, the local r-triplets are now using integers. These integer values represent the size of their associated pairing partial sets, and will help the agents to determine the sizes of the different overall pairing partial sets. Once the sizes of the different overall pairing partial sets have been determined, the agents can find which overall pairing partial set contains τ examples or more, and therefore associate an overall pairing relation to C_i and C_j which acknowledge an error degree τ. These local r-triplets of our error-tolerant model are presented in Definition 51 below:

Definition 51 (Local R-Triplet). *Let ev be the function defined in Definition 20 and $h(U_x)$ the function that, for every pairing partial set U_x, yields:*

$$h(U_x) = \begin{cases} \tau, & \text{if} |U_x| \geq \tau. \\ |U_x|, & \text{otherwise.} \end{cases} \tag{6.1}$$

The function $r_l : \mathbb{X} \times \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{N}^3$, with $\mathbb{X}$ the domain of concepts and $\mathbb{U}$ the domain of all local contexts, is a function that for each pair of concepts C_i,C_j and for a given context U yields a triplet $r_l(C_i, C_j, U) = (i_{-1}, i_0, i_1)$, called local r-triplet, such that for $x \in \{-1, 0, 1\}$:

$$i_x = ev(x, C_i, C_j, U, h)$$

Loose R-Triplets

In the error-free model, the agents could combine their local r-triplets to find an overall r-triplet. From the overall r-triplet, the agents could know which overall pairing partial sets were empty and which were not. In a model that assumes a degree of error, the agents do not aim to find which overall pairing partial sets are empty but which overall pairing partial sets contain less than τ examples. From their two local r-triplets, the agents can infer some values of the overall r-triplet, but some other values will remain unknown for the moment. An overall r-triplet that contains unknown values is an overall *loose* r-triplet. Loose r-triplets cannot be used to find the overall pairing relation between C_i and C_j directly, but remains a good starting point. The overall loose r-triplets of two local r-triplets is defined below:

Definition 52 (Loose R-Triplet)**.** *The function $r_{ol} : \mathbb{N}^3 \times \mathbb{N}^3 \rightarrow (\mathbb{N} \cup \{?\})^3$ is a function that for each pair of local r-triplet r_1, r_2 yields a triplet $r_{ol}(r_1, r_2) = (i_{-1}, i_0, i_1)$ called loose r-triplet, such that for $x \in \{-1, 0, 1\}$:*

$$i_x = \begin{cases} \tau, & \textit{if } r_1[x] = \tau \textit{ or } r_2[x] = \tau. \\ 0, & \textit{if } r_1[x] + r_2[x] < \tau. \\ ? & \textit{otherwise.} \end{cases} \tag{6.2}$$

Given two concepts C_1 and C_2, and two local contexts U_1 and U_2, the notation $r_{ol}(C_1, C_2, U_O)$ refers to the loose r-triplet $r_{ol}(r_l(C_1, C_2, U_1), r_l(C_1, C_2, U_2))$.

Loose r-triplets are a good intermediate step to find the overall r-triplet because a loose r-triplet already gives some partial information on which overall pairing partial sets contain more than τ examples.

Conjecture 5 (Loose R-Triplet Usefulness)**.** *Let A_1 and A_2 be two agents, A_1 partitioning the context U_1 in the contrast set $K_1 = (U_1, S_1)$ and A_2 partitioning the context U_2 in the contrast set $K_2 = (U_2, S_2)$. Let $C_i \in S_1$ and $C_j \in S_2$ be two concepts, and $r = r_{ol}(C_i, C_j, U_O)$ their loose r-triplet in the overall context. Let ev be the function defined in 20. For $x \in \{-1, 0, 1\}$:*

- $r[x] = 0 \implies |U_O(x, C_i, C_j)| < \tau$
- $r[x] = \tau \implies |U_O(x, C_i, C_j)| \geq \tau$

We have not finalized the formal proof of this conjecture, but will explain the main ideas behind it. The schema of the proof would be the following: we prove that if an element $r[x]$ of the loose r-triplet is equal to zero, then the corresponding values $r_1[x]$ and $r_2[x]$ have, according to Definition 52, a sum that is lower than τ. We prove that if the sum of $r_1[x]$ and $r_2[x]$ is lower than τ then the number of examples in the union of their corresponding pairing partial sets $U_1(x, C_i, C_j)$ and $U_2(x, C_i, C_j)$ is less than τ. Applying Conjecture 2, we may prove that if $r[x]$ is equal to zero, then its corresponding overall partial set $U_O(x, C_i, C_j)$ contains less than τ examples.

We also prove that if $r[x]$ is equal to τ then, according to Definition 52, one of the two values $r_1[x]$ or $r_2[x]$ is higher than τ. We prove that if $r_1[x]$ or $r_2[x]$ is higher than τ, then one of their corresponding pairing partial sets $U_1(x, C_i, C_j)$ or $U_2(x, C_i, C_j)$ contains at least τ examples and therefore their union also contains at least τ examples. Applying Conjecture 2, we prove that if $r[x]$ is equal to τ, then its corresponding overall partial set $U_O(x, C_i, C_j)$ contains at least τ examples.

Therefore, we may prove that if $r[x]$ is equal to zero then the corresponding overall pairing partial set $U_O(x, C_i, C_j)$ contains less than τ examples, and that if $r[x]$ is equal to τ then the corresponding overall pairing partial set $U_O(x, C_i, C_j)$ contains at least τ examples. □

Secured R-Triplets

Once the agents have computed the loose r-triplets, they need to attribute an integer to the elements of unknown values. The next step for the agent is to produce a r-triplet similar to a loose r-triplet, but with each unknown value replaced by 0 if its corresponding overall pairing partial set contains less than τ examples and replaced by τ otherwise. A such triplet is called an overall *secured* r-triplet.

Definition 53 (Secured R-Triplet)**.** *Let ev be the function defined in Definition 20 and $p(U_x)$ the function that, for each pairing partial set U_x, yields the value:*

$$p(U_x) = \begin{cases} \tau, & if |U_x| \geq \tau. \\ 0, & otherwise. \end{cases} \tag{6.3}$$

The function $r_{os} : \mathbb{X} \times \mathbb{X} \times \mathbb{U} \to \mathbb{N}^3$, with $\mathbb{X}$ the domain of concepts and $\mathbb{U}$ the domain of all local contexts, is a function that for each pair of concepts C_i,C_j and for a given context U yields a triplet $r_{os}(C_i, C_j, U) = (i_{-1}, i_0, i_1)$ called secured r-triplet, such that for $x \in \{-1, 0, 1\}$:

$$i_x = ev(x, C_i, C_j, U, p)$$

Finding the overall r-triplet of two concepts C_i and C_j with Definition 53 alone would be complicated, as the agents would need to have access to the three overall pairing partial sets of C_i and C_j. However, it is possible to obtain the overall secured r-triplet of C_i and C_j with potentially much less information exchanged by using the overall loose r-triplet of $r_l(C_i, C_j, U_1)$ and $r_l(C_i, C_j, U_2)$. This is due to the fact that, according to Conjecture 5, given:

- the loose r-triplet $r = r_{ol}(C_i, C_j, U_O)$ and
- the secured r-triplet $r' = r_{os}(C_i, C_j, U_O)$,

the known value (value different from ?) $r[x]$ of the loose r-triplet r is equal to the value $r'[x]$ of r', as these two values have the same definition. Since computing an overall loose r-triplet is not costly for the agents (they only to exchange three integers each), the agents can look for the overall loose r-triplet of two concepts and then replace each of its unknown values by the value of same index from the overall secured r-triplet. If the agents follow this method, they do not have to compute the whole secured r-triplet but only some of its values.

Tight R-Triplets

According to Definition 53, the agents can find each unknown value $r[x]$ from the loose r-triplet $r = r_{ol}(C_i, C_j, U_O)$ by accessing the corresponding pairing partial set $U_O(x, C_i, C_j)$. According to Conjecture 1:

$$U_O(x, C_i, C_j) = U_1(x, C_i, C_j) \cup U_2(x, C_i, C_j).$$

The issue is that each local pairing partial set $U_k(x, C_i, C_j)$ is a subset of the local context U_k of the agent A_k, and U_k is supposed to be accessed only by A_k. However, we saw in Section 5.1.3 that an agent can choose to share examples with the other, by sending a message *Examples()*. It is therefore possible for an agent A_k to access an overall pairing partial set $U_O(x, C_i, C_j)$, as long as the other agent A_{-k} sends the examples $U_{-k}(x, C_i, C_j)$ to A_k. Since our agents are cooperating,

A_{-k} will even take the initiative to send these examples to A_k if it realizes that A_k needs them as we will later see in Section 7.4.1.

Upon receiving complementary information on the overall context from the other agent, represented by an extended local context U'_k containing new examples, an agent A_k can try to replace the unknown values $r[a] =?, \ldots, r[c] =?$ of the loose r-triplet r by new known values. The resulting triplet is called a *tight* r-triplet of r. The notion of tight r-triplet is defined below in Definition 54. Under the right circumstances, tightening the loose triplet $r_{ol}(C_i, C_j, U_O)$ can make it equivalent to the secured r-triplet $r_{os}(C_i, C_j, U_O)$.

Definition 54 (Tight R-Triplet). *The function $r_{os} : \mathbb{N}^3 \times \mathbb{U} \to \mathbb{N}^3$, with $\mathbb{U}$ being the domain of all sets of examples, is a function that, for a r-triplet r and a given set of example U, yields a triplet $r_{ot}(r, U) = (i_{-1}, i_0, i_1)$ called tight r-triplet, such that for $x \in \{-1, 0, 1\}$:*

$$i_x = \begin{cases} r[x], & \textit{if } r[x] \neq ?. \\ \tau, & \textit{if } r[x] =? \textit{ and } U(x, C_i, C_j) \geq \tau. \\ 0, & \textit{if } r[x] =? \textit{ and } U(x, C_i, C_j) < \tau. \end{cases} \tag{6.4}$$

Given two concepts C_1 and C_2, we also use the alternative notation:

$$r_{ot}(C_1, C_2, U_k \cup U) = r_{ot}(r_{ol}(C_1, C_2, U_O), U).$$

Conjecture 6 below gives an idea of how the agents can select which local pairing partial sets to exchange in order to secure a specific value $r[x]$ of an overall loose r-triplet r. Ideally, the agent A_k with the less examples in its pairing partial set $U_k(x, C_i, C_j)$ sends it to A_{-k} in order for A_{-k} to tighten r with an extended context $U_{-k} \cup U_k(x, C_i, C_j)$ which is equivalent to $U_O(x, C_i, C_j)$, therefore securing the value $r[x]$ in the resulting tight triplet. The agents can determine which of the two of them has the least examples in its local pairing partial set by checking the two local r-triplets r_1 and r_2 that were used to compute $r = r_{ol}(r_1, r_2)$. In the situation where $r_1[x] = r_2[x]$ and the two local pairing partial sets have the same size, the agents randomly choose which of the two of them will send its local pairing partial set, and which one will receive it to tighten the loose r-triplet.

Conjecture 6 (Tight R-Triplet Usefulness). *Let A_1 and A_2 be two agents, A_1 partitioning the context U_1 in the contrast set $K_1 = (U_1, S_1)$ and A_2 partitioning the context U_2 in the contrast set $K_2 = (U_2, S_2)$. Let $C_i \in S_1$ and $C_j \in S_2$ be two concepts, x and integer from $\{-1, 0, 1\}$. Let U^* be a context such that $U^* \supseteq U_1(x, C_i, C_j) \cup U_2(x, C_i, C_j)$. Let:*

- *r be the loose r-triplet $r_{ol}(C_i, C_j, U_O)$,*
- *r' be the tight r-triplet $r_{ot}(r, U^*)$, and*
- *r'' be the secured r-triplet $r_{os}(C_i, C_j, U_O)$.*

In these conditions, $r'[x] = r''[x]$ holds.

We have not finalized the formal proof of this conjecture, but will explain the main ideas behind it. The schema of the proof would be the following: Conjecture 5 states that if the element $r[x]$ of the loose r-triplet is known and equal to zero, then its corresponding overall pairing partial set $U_O(x, C_i, C_j)$ contains less than τ elements. We prove that according to Definition 53, if the overall pairing partial set $U_O(x, C_i, C_j)$ contains less than τ examples, then the element $r''[x]$ of the secured r-triplet is equal to zero. We prove that, according to Definition 54, if $r[x]$ is equal to zero then so is $r'[x]$. Therefore, we may prove that $r[x]$ being equal to zero means that $r'[x]$ and $r''[x]$ are both equals.

Conjecture 5 also states that if the element $r[x]$ of the loose r-triplet is known and equal to τ, then its corresponding overall pairing partial set $U_O(x, C_i, C_j)$ contains at least τ elements. We prove that, according to Definition 53, if the pairing partial set $U_O(x, C_i, C_j)$ contains at least τ examples, then the element $r''[x]$ of the secured r-triplet is equal to τ. We prove that according to Definition 54, if $r[x]$ is equal to τ then so is $r'[x]$. Therefore, we may prove that $r[x]$ being equal to τ means that $r'[x]$ and $r''[x]$ are both equals.

We recall, from the proof schema of Conjecture 1, that each pairing partial set $U_k(x, C_i, C_j)$ of a local context U_k is the set of examples from U_k that verifies a certain predicate Φ_x. We should also recall that the examples from the union of $U_1(x, C_i, C_j)$ and $U_2(x, C_i, C_j)$ are the examples that verify a same predicate Φ. Next we would prove the following equality:

$$U^*(x, C_i, C_j) = \{e \in \{e' \in (U_1(x, C_i, C_j) \cup U_2(x, C_i, C_j)) | \Phi_x(e)\} | \Phi_x(e)\}$$

Then we would prove the following equality:

$$U^*(x, C_i, C_j) = \{e \in (U_1(x, C_i, C_j) \cup U_2(x, C_i, C_j)) | \Phi_x(e)\}$$

Applying Conjecture 1, we prove that the set of the examples from the union of the pairing partial sets $U_1(x, C_i, C_j)$ and $U_2(x, C_i, C_j)$ that verify Φ is the overall pairing partial set $U_O(x, C_i, C_j)$, and therefore that $U_O(x, C_i, C_j) = U^*(x, C_i, C_j)$. We prove that according to Definition 53, $r"[x]$ is equal to 0 if $U^*(x, C_i, C_j)$ contains less than τ examples and $r"[x]$ is equal to τ if $U^*(x, C_i, C_j)$ contains at least τ examples. We may prove that, according to Definition 54, $r'[x]$ is equal to 0 if $U^*(x, C_i, C_j)$ contains less than τ examples and $r'[x]$ is equal to τ if $U^*(x, C_i, C_j)$ contains at least τ examples. Therefore, we may prove that $r'[x] = r''[x]$. □

We presented how to find the secure r-triplet $r' = r_{os}(C_i, C_j, U_O)$ from a loose r-triplet $r = r_{ol}(C_i, C_j, U_O)$ if one of the values from r is unknown. However, this method cannot be applied to the situations where two values or more from r are unknown values. For instance, let's consider these two local r-triplets:

- $r_1 = r_l(C_i, C_j, U_1) = (9, 8, 10)$, and
- $r_2 = r_l(C_i, C_j, U_2) = (8, 9, 10)$.

Let's now assume that $\tau = 10$. From the two triplets r_1 and r_2, A_1 and A_2 can both compute the loose r-triplet $r' = r_{ol}(C_i, C_j, U_O) = (?\,, ?\,, \tau)$. In this situation, A_1 will send its local pairing partial set $U_1(0, C_i, C_j)$ to A_2 and A_2 will send its local paring partial set $U_2(-1, C_i, C_j)$ to A_1 because:

1. $|U_1(0, C_i, C_j)| < |U_2(0, C_i, C_j)|$, and
2. $|U_1(-1, C_i, C_j)| > |U_2(-1, C_i, C_j)|$.

Now, if the overall pairing partial sets have the following sizes:

- $|U_O(-1, C_i, C_j)| = 11$, and
- $|U_O(0, C_i, C_j)| = 9$,

then, the tight r-triplets that the agents compute will be:

- $r''_1 = r_{ot}(C_i, C_j, U_1 \cup U_2(-1, C_i, C_j)) = (\tau, 0, \tau)$ for A_1, and

- $r_2'' = r_{ot}(C_i, C_j, U_2 \cup U_1(0, C_i, C_j)) = (0, 0, \tau)$ for A_2.

The agent A_2 has correctly secured the value i_0 of r' in r_2'', but failed to correctly secure the value i_{-1} of the same loose r-triplet. The tight r-triplet r_2'' is different from the overall secured r-triplet $r_{os} = (\tau, 0, \tau)$. If there can be more than one unknown value in a loose r-triplet r, the agents should use another method than only computing a tight r-triplet of r if they want to find the overall secured r-triplet. By using Conjecture 7 presented below, the agents can combine two tight r-triplets in order to find an overall secure r-triplet.

Conjecture 7 (Combining Tight R-Triplets). *Let A_1 and A_2 be two agents, A_1 partitioning the context U_1 in the contrast set $K_1 = (U_1, S_1)$ and A_2 partitioning the context U_2 in the contrast set $K_2 = (U_2, S_2)$. Let $C_i \in S_1$ and $C_j \in S_2$ be two concepts. Let $r = r_{ol}(C_i, C_j, U_O)$ be a loose r-triplet and $A, B \subseteq \{-1, 0, 1\}$. Let r_1 and r_2 be two tight r-triplets such that:*

- $r_1 = r_{ot}(r, U_1 \cup (\bigcup_{x \in A} U_2(x, C_i, C_j)))$, *and*
- $r_2 = r_{ot}(r, U_2 \cup (\bigcup_{x \in B} U_1(x, C_i, C_j)))$.

Then, let $r' = (i_{-1}, i_0, i_1)$ be the triplet that, for all $x \in \{-1, 0, 1\}$, carries the value:

$$i_x = \begin{cases} \tau & \textit{if } r_1[x] = \tau \textit{ or } r_2[x] = \tau \\ 0 & \textit{otherwise} \end{cases}$$

and r'' be the secured r-triplet $r_{os}(C_1, C_2, U_O)$. In these conditions, the following holds:

$$A \cup B = \{x \in \{-1, 0, 1\} | r[x] =?\} \Leftrightarrow (r' = r'').$$

We have not finalized the formal proof of this conjecture, but will explain the main ideas behind it. We prove that x belongs to either the set A or the set B. We prove that x belonging to A is equivalent to the pairing partial sets $U_1(x, C_i, C_j)$ and $U_2(x, C_i, C_j)$ being subsets of the set $U_1 \cup (\bigcup_{x \in A} U_2(x, C_i, C_j)))$, that we will write U_A. We will prove that the union of $U_1(x, C_i, C_j)$ and $U_2(x, C_i, C_j)$ is also a subset of U_A. We will prove that, according to Conjecture 6, the union of the pairing partial sets $U_1(x, C_i, C_j)$ and $U_2(x, C_i, C_j)$ being a subset of U_A is equivalent to $r_1[x] = r''[x]$. Therefore, we prove that x belonging to A is equivalent to $r_1[x] = r''[x]$. We would use the same reasoning to prove that x belonging to B is equivalent to $r_2[x] = r''[x]$.

We will prove that x belonging to A or B is equivalent to either $r_1[x]$ or $r_2[x]$ being equal to $r''[x]$. We will prove that x belonging to $A \cup B$ is equivalent to $r''[x] = \tau$ if and only if $r_1[x] = \tau$ or $r_2[x] = \tau$. By definition, x belonging to $A \cup B$ and the value of either $r_1[x]$ or $r_2[x]$ being equal to τ, is equivalent to $r'[x] = \tau$. Therefore, since r_1, r_2 and r' can only take values from $\{0, \tau\}$, we may prove that $x \in A \cup B$ is equivalent to $r'[x] = r''[x]$. □

In these conditions, if one agent A_k makes sure, for each unknown value $r[x]$ of a loose r-triplet r, to send its pairing partial set $U_k(x, C_i, C_j)$ to A_{-k} if $U_k(x, C_i, C_j)$ contains less examples than $U_{-k}(x, C_i, C_j)$, and if the other agent A_{-k} makes sure to send its pairing partial set $U_{-k}(x, C_i, C_j)$ to A_{-k} if $U_{-k}(x, C_i, C_j)$ contains at most the same number of examples as $U_k(x, C_i, C_j)$, then according to Conjecture 7 the two agents A_k and A_{-k} can each compute a tight r-triplet that can be combined into the corresponding overall secured r-triplet.

Binarized R-Triplet

Once an agent has found the overall secured r-triplet r for the pair of concepts C_i, C_j, it still needs to transform this triplet of integers into a triplet of Boolean values, as a pairing relation is defined in Definition 22 by a triplet of Boolean values and not a triplet of integers. We call this final triplet of Boolean values the overall *binarized* r-triplet.

Definition 55 (Binarized Overall R-Triplet)**.** *The function $r_{ob} : \mathbb{N}^3 \to \mathbb{N}^3$ is a function that, for a triplet (i'_{-1}, i'_0, i'_1), yields a binarized triplet (i_{-1}, i_0, i_1) defined as follows:*

$$i_x = \begin{cases} 0, & \text{if } i'_x < \tau \\ 1, & \text{otherwise} \end{cases}$$

The binarized triplet of a secured r-triplet $r_{os}(C_i, C_j, U_O)$ can be used together with the pairing function presented in Definition 22 to find the overall pairing relation $C_i r_{UO} C_j$ between two concepts C_i and C_j.

6.4 Effect on Concept Creation through Argumentation

In Section 5.1.3, we explained that the creation of a new concept C_i always start with the choice of a subset U_i^+ of the overall context. Until now, when the agents are trying to learn an intensional definition I_i to subsume U_i^+, they were not expected to make any type-one or type-two error. This means that I_i was supposed to subsume all the elements of U_i^+, and none from the rest of the overall context of the two agents. However, with the introduction of an error degree, the agents are now allowed to have I_i subsuming some examples that are not from U_i^+, and to not subsume some other examples that are not from U_i^+. We write:

- FP the examples from the set U_i^+ that are subsumed by I_i, as they are false positives.
- FN the examples from the set U_i^- that are not subsumed by I_i, as they are false negatives.

Assuming a degree of error τ means that we are expecting the number of false positives and negatives to be less than τ in total. Since the set of false positives and the set of false negatives are disjoint, their union verifies $|FP \cup FN| = |FP| + |FN|$. Therefore, FP and FN should verify:

$$|FP| + |FN| < \tau_E.$$

The agents should therefore ensure that the sum of the false positives and the false negatives is less than τ by only accepting intensional definitions that have their false positives and negatives verifying:

$$(1)\ |FP| < \tau/2, \text{ and } |FN| < 2/\tau.$$

In these circumstances, the sum of the false positives and negatives will always be less than τ. However, an agent A_k which has a context U_k cannot access FP or FN —that have elements in both contexts– and therefore cannot verify (1). However, A_k can verify that:

$$(2)\ |FP \cap U_k| < 4/\tau, \text{ and } |FN \cap U_k| < 4/\tau.$$

If both A_1 and A_2 verify that (2) is true, then A_1 and A_2 have verified that $|(FP \cap U_1) \cup (FP \cap U_2)| < \tau/2$ and $|(FN \cap U_1) \cup (FN \cap U_2)| < \tau/2$. According to Definition 24, the overall context is the union of the two local contexts. Therefore:

- $(FP \cap U_1) \cup (FP \cap U_2) = FP$, and
- $(FP \cap U_1) \cup (FP \cap U_2) = FN$.

This means that if each agent A_k only accept a set of generalizations I_i as the intensional definition of the new concept C_i if I_i has less than $\tau/4$ false positives and $\tau/4$ false negatives in U_k, the two agents A_1 and A_2 can be sure that the intensional definition I_i commits less than τ classification error over U_O.

The agents also need to make sure that there are at least τ examples in the adjunct set $Adj(I_i, U_O)$. Since I_i subsumes the set of positive examples plus the false positives and minus the false negatives, in order to have at least τ examples in $Adj(I_i, U_O)$ the intensional definition I_i should verify:

$$|U_i^+| + FP - FN > \tau$$

This means that the number of false negatives FN should verify $|U_i^+| + FP - \tau > FN$. In the worst case scenario, there are no false positives and the set of positive examples minus the error threshold should be more than the false negatives. This means that the intensional definition I_i should never have more than $|U_i^+| - \tau$ false negatives (3). The agents have not access to either U_i^+ or FN and therefore cannot verify (3) individually. However, each agent A_k can verify that:

$$(4)\ |FN \cap U_k| < |U_i^+ \cap U_k| - \tau/2.$$

If both A_1 and A_2 verify (4), then the two agents can be sure that $|FN| < |U_i^+| - \tau$ and therefore that the adjunct set $Adj(I_i, U_O)$ has enough examples to allow C_i to be a concept.

6.5 Effect on the Transitivity of the Equivalence Pairing Relation

In a model that assumes a degree of error, the pairing relation of equivalence is not transitive anymore. Figure 6.2 illustrates a situation where, for $\tau = 4$, two concepts C_1 and C_2 share more than τ examples, with $C_1 - C_2$ and $C_2 - C_1$ having both less than τ examples, meaning that C_1 and C_2 are equivalent in U. The remark is the same for C_2 and the third concept C_3. However, C_1 and C_3 share less than τ examples and therefore are not equivalent. This means that while $C_1 \equiv C_2$ and $C_2 \equiv C_3$, $C_1 \not\equiv C_3$. Clearly, the pairing relation $\equiv$ is not transitive in $\{C_1, C_2, C_3\}$.

However, the transitivity of the equivalence relation is needed in our model. For instance, two disjoint concepts C_1 and C_2 from a same contrast set that are both equivalent to a third concept C_3 from another contrast set can cause a dead loop in the argumentation. If C_1 and C_2 share the same sign, they are homonyms and cause a homonymy disagreement. If C_1 and C_2 have different signs, then at least one of them has a different sign than C_3 and therefore cause a synonymy disagreement. Therefore C_1 and C_2 would need to have the same sign and different signs at the same time, making the mutual intelligibility impossible to reach. The only solution to this problem is to have the transitivity of the equivalence relation enforced by the agents. When the agents detect two non-equivalent concepts that are both equivalent to a third concept, the agents remove one of the two non-equivalent concepts from the argumentation.

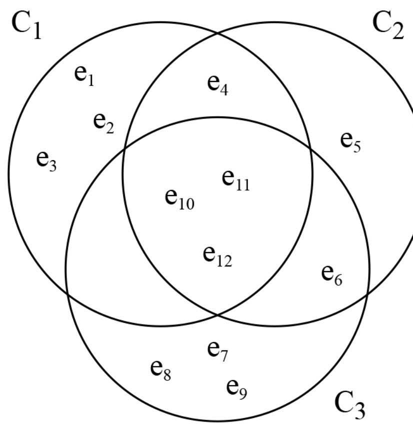


Figure 6.2: An example of transitivity violation. In this figure, twelve examples $e_1, \ldots, e_{12}$ of a are placed in a Venn diagram representing three concepts. The examples subsumed by a concept are placed within its circle.

6.6 Conclusion

This Chapter presents how the approach presented in Chapter 4, and the model presented in Chapter 5, can be adapted to an error-tolerant model. The adaptation is based on the introduction of local r-triplets, loose r-triplets, tight r-triplets and secured r-triplets. We present how theses triplets can be used to infer the overall pairing relations in an error tolerant model. This inference is based on the three conjectures presented in this chapter and Conjectures 1 and 4. Since this inference method is incorporated in our implementation, and our experimental evaluation in Chapter 10 shows that the agents exchanging the r-triplets achieve the desired resolution of disagreements, we can conclude that the conjectures usefulness is supported empirically.

Chapter 7

Systematic Strategy to Mutual Intelligibility

7.1 Introduction

The systematic strategy is our first approach to reach mutual intelligibility, using the model presented in the precedent chapters. The systematic strategy consists into systematically searching for synchronic disagreements between the two agents, and resolve these disagreements once they are all listed. The disagreements are resolved according to the methods presented in Section 5.5 of Chapter 5. As we mentioned in Section 5.5, resolving a semantic disagreement can sometimes create new synchronic disagreements. For this reason, the systematic strategy also look for new disagreements each time that a disagreement has been resolved. Once the systematic strategy has brought the agents into mutual intelligibility, the systematic strategy finishes by changing the vocabulary of the two agents, reusing the past vocabulary of their initial contrast sets. This additional but optional phase is cosmetic, allowing the new contrast sets to use real signs instead of generated ones. The strategy that is described below works for models that admit a degree of error τ.

7.2 Structure of an Argumentation Adopting a Systematic Strategy

The systematic strategy is characterized by its linear structure. While its structure includes a loop, the strategy has a clear beginning and an end point, which differs from the "on-demand" design of the lazy strategy. The argumentation strategy is structure in four main phases: *Start*, *Evaluation*, *Resolve Disagreements* and the optional *Update Vocabulary*. The agents follow these phases, cycling between Phase 2 and Phase 3 until reaching mutual intelligibility. Each phase is divided into main steps. A step corresponds to a short term objective in term of argumentation for the agents. For instance, the Evaluation Phase contains the step *Identify Pairing Relations*. Figure 7.1 represents the four different phases and their respective steps. Each step is also divided into *states*. The states, already mentioned in Section 5.1.2, are the different algorithms that an agent can follow during one of its turns. This algorithm always end by sending back the token to the other agent, so this action is not written in the state presentations listed in the sections below. If one state $\#A$ has only one possible next state $\#B$, the second-to-last action that an agent takes in the state $\#A$ is to set up its next state to $\#B$. Before selecting its next state, an agent always empty its mailbox unless specified otherwise.

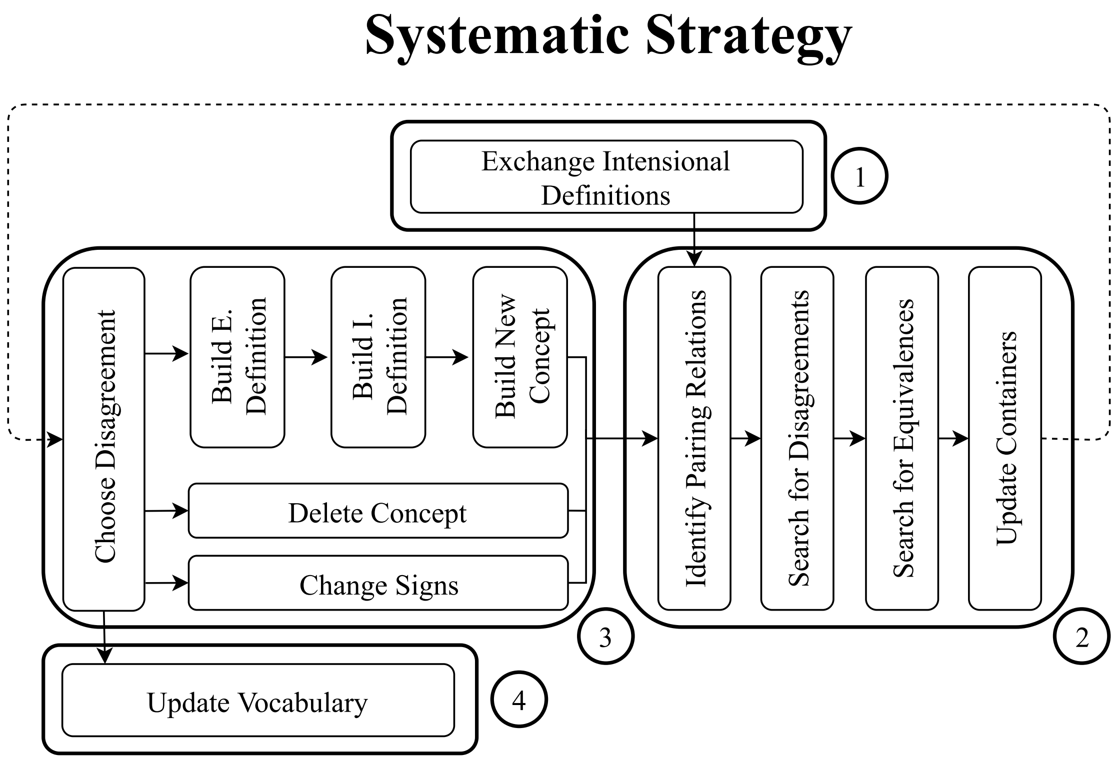


Figure 7.1: Diagram of the four main phases of the systematic strategy for argumentation (bold rectangles), with their respective steps (thin rectangles). Each step is linked to its possible next steps by an arrow. The dashed-line arrow symbolizes the main loop of our argumentation strategy, between the phases 2 and 3.

7.3 Phase 1: Start

The start of our argumentation strategy is an introduction of the agents in the cycle that will later take place between Phases 2 and 3. In this phase, the agents exchange all their intensional definitions and prepare for argumentation v. The State 1 *Send Intensional Definition* is the initial state of the agents, and defines the first actions that the agents will take during their first turn.

7.3.1 Step 1: Exchange Intensional Definitions

This step is the unique step of Phase 1. During its first state, the agents exchange all their intensional definitions. During the second state, each agent creates a hypothesis using the intensional definitions that it received from the other agent. At the end of this state, each agent has a hypothesis emulating all the concepts from their interlocutor's contrast set. They can start identifying disagreements.

State 1: Send Intensional Definition

- **Input Messages:** Since the agent will only be in this state during its first turn, no message is expected at the beginning of this turn.
- **Output Messages:** *Assert, Check-Self*

- **Next Possible State(s):** State 2

Upon receiving the token the agent A_k creates a new contrast set $K = \{S_{K,k}, U_k\}$, copy of A_k's initial contrast set K_i. For each concept $C_i \in S_{K,k}$, A_k:

1. adds C_i to its list of newly created concepts Add_k,
2. creates a message $m = Assert\#2(s(C_i),\ id(C_i),\ I(C_i))$,
3. sends m to the other agent

Additionally, A_k sends a *Self-Check*#3() message to A_{-k} as a reminder that the first time the agents go through Phase 2 they should also evaluate the pairing relations $R(Add_K \times Add_K, U_O)$ and $R(Add_H \times Add_H, U_O)$, in order to look for Self-Disagreements.

State 2: Receive Intensional Definition

- **Input Messages:** *Assert*
- **Next Possible State(s):** State 3

The agent A_k creates a new hypothesis $H = \{S'_{H,k} =, U_k\}$. For each message $Assert(s,\ id,\ I))$ in its mailbox, A_k adds a new concept $C'_i = \langle s, I, Adj(I, U_k)\rangle$ to $S_{H,k}$ such that $id(C'_i) = id$ and adds C'_i to its list Add_H of concepts that have been newly created by A_{-k}.

7.4 Phase 2: Evaluation

During the last phase – either phase 1 or 3, the contrast sets of the agents have been modified. Concepts might have been added or deleted. For this reason, the agents need to update their hypotheses and to (re-)evaluate the pairing relations between each of their concepts, then detect the eventual disagreements, and list these disagreements. Additionally, the agents need to check that the transitivity of the equivalence relation is still respected within both contrast sets, and take measures if this is not the case. Once the evaluations are done and the containers updated, the agents are ready to resolve a new disagreement if there is any left.

7.4.1 Step 2: Identify Pairing Relations

This step is the first of Phase 2. During this step, the agents determine the pairing relation between each of their concepts. Since we determined that the synchronic disagreements are caused by specific pairing relations in Section 4.5.3, knowing the overall pairing relations of each pair of concepts allows the agent to list their disagreements. Without listing their disagreements, the agents logically cannot resolve them. Multiple things are listed during this step: overall r-triplets, overall pairing relations, and hierarchies (which concept from a hypo/hypernymy relation is the hyponym and which is the hypernym). If this is the first time that the agents enter Phase 2, all their concepts should be in there respective lists Add_K and Add_H, allowing the agents to check all the initial pairing relations in the argumentation.

State 3: Send Local R-Triplets

- **Input Messages:** *Self-Check*
- **Output Messages:** *Evaluation*
- **Next Possible State(s):** State 4

Upon receiving the token, A_k checks whether or not it received a *Self-Check* message. Then, A_k computes the set of local r-triplets T that is equal to:

- $T(Add_K \times S_{H,k}, U_k) \cup T(S_{K,k} \times Add_H, U_k) \cup T(Add_K \times Add_H, U_k)$,
- and additionally $T(Add_K \times Add_K,\ U_k) \cup T(Add_H \times Add_H,\ U_k)$ if a *Self-Check* message has been received.

For each local r-triplet $r_l = r(C_i, C_j, U_k) \in T$, A_k sends a message *Evaluation*#4($id(C_i)$, $id(C_j)$, r_l) to A_{-k}. Then, A_k adds to its containers the concepts that have been eventually created during the previous phase, and that are stored in the lists Add_K and Add_H: for each concept C in the list Add_K, A_k adds C to $S_{K,k}$, then for each concept C' in the list Add_H, A_k adds C' to $S_{H,k}$.

State 4: Send Loose R-Triplets

- **Input Messages:** *Evaluation, Seize*
- **Output Messages:** *Evaluation, Examples*
- **Next Possible State(s):** State 5

The agent A_k check if it has received a *Seize* message, indicating that the other agent will be in charge of sending the same-size pairing partial sets associated to unknown values in the loose r-triplets. Then, for each message *Evaluation*($id(C_i)$, $id(C_j)$, r) in its mailbox, A_k computes the loose r-triplet $r' = r_{ol}(C_i, C_j, U_O)$ using the method described in Section 6.3.2. For each value $r'[x]$ of the r-triplet r', if $r'[x]$ is unknown and either:

1. $|U_k(x, C_i, C_j)| < |U_{-k}(x, C_i, C_j)|$, or
2. $|U_k(x, C_i, C_j)| = |U_{-k}(x, C_i, C_j)|$, without the other agent having seized the computation of the overall r-triplet yet,

A_k sends the set of examples $U = U_k(x, C_i, C_j)$ to the other agent through a message *Examples*#5(U). Doing so, A_k allows the other agent A_{-k} to secure the value $r'[x]$ by making possible for A_{-k} to build a super-set of the overall pairing partial set $U_O(x, C_i, C_j)$ and computing a tight r-triplet. A_k knows that this option is better than A_{-k} sending $U_{-k}(x, C_i, C_j)$ to A_k, since there are more examples in $U_{-k}(x, C_i, C_j)$. If $U_k(x, C_i, C_j)$ and $U_{-k}(x, C_i, C_j)$ have the same size, the first agent A_l to enter State 4 will send the examples $U_l(x, C_i, C_j)$ to the other agent A_{-l}, along with a message *Seize*#() to A_{-l} in order to prevent A_{-l} to do the same. Indeed, since A_{-l} will be able to secure $r'[x]$, sending $U_{-l}(x, C_i, C_j)$ to A_l for A_l to secure $r'[x]$ too would be redundant.

This method allows the agents to only exchange the smallest of their pairing partial sets, and to deal with situations where both pairing partial sets have the same size. Thanks to the integer-triplets, the agents can know how many examples there are in the other agent's pairing partial sets, as the undefined values can only appear if there are less than τ examples in both pairing partial sets. Exchanging these examples is necessary to compute the undefined values, as explained in Section 6.3.2. After sending all the necessary pairing partial sets, the agent A_k sends an *Evaluation*#5($id(C_i)$, $id(C_j)$, r') to the other agent.

State 5: Send Tight R-Triplets

- **Input Messages:** *Evaluation, Examples*
- **Output Messages:** *Evaluation*
- **Next Possible State(s):** State 6

During this state, the agents will each secure one split the unknown values from each loose triplet that has been computed during State 4. The arrangements made in State 4 ensures that the values that will not be secured by one agent will be secured by the other during this state, and that each unknown value is secured by only one agent. The agent A_k starts its turn by adding all the examples that it has received from the *Example* messages in its mailbox, to all of its currents contexts – which means both $U_{K,k}$ and $U_{H,k}$. Once the examples have been added, A_k computes the tight r-triplet $r' = r_{ot}(C_i, C_j, U_k)$ for each message $Evaluation(id(C_i),\ id(C_j),\ r)$ that it has received in its mailbox, and sends r' to A_{-k} through a message $Evaluation\#6(id(C_i),\ id(C_j),\ r')$.

State 6: Compute Secured R-Triplets

- **Input Messages:** *Evaluation*
- **Output Messages:** *Relation*
- **Next Possible State(s):** State 7

For each message $Evaluation(id(C_i),\ id(C_j),\ r')$ in its mailbox, the agent A_k combines r' with the tight r-triplet $r = r_{ot}(C_i, C_j, U_k)$. Since for each unknown value i_n of the loose r-triplet from which r and r' are derived, the associated pairing partial set $U_O(i_n, r)$ is now a subset of one of the two local contexts U_1 or U_2, then combining the two r-triplets r and r' according to the method described in Section 6.3.2 produces a new r-triplet $r'' = r_{os}(C_i, C_j, U_O)$. A_k memorizes the binarized r-triplet $r_b(r'')$ in a list RT, and uses $r_b(r'')$ to find the overall pairing relation between C_i and C_j using Definition 22. If the overall pairing relation that A_k finds is a hypo/hyperonymy, A_k also keeps in memory which concept is the hyponym and which is the hypernym in the list Hh. Then, A_k sends a message $Relation\#7(id(C_i),id(C_j),C_i r_{UO} C_j)$ to A_{-k}, so A_{-k} can list the disagreements during the next turn.

7.4.2 Step 3: Search for Disagreements

The third step of Phase 2 has only one state. After having determined the overall r-triplets and overall pairing relations between all the concepts newly involved in the argumentation, the agents can check whether each overall pairing relation is causing a disagreement. Such overall pairing relations need to be listed in order for their associated synchronic disagreements to be resolved later during the third Phase.

State 7: Find Disagreements

- **Input Messages:** *Relation*
- **Next Possible State(s):** State 8

For each message $Relation(id(C_i),id(C_j),C_i r_{UO} C_j)$ in its mailbox, the agent A_k check whether are not the relation $r = C_i r_{UO} C_j$ is causing a synchronic disagreement, as explained in Section 4.5.3. If the relation is causing a disagreement, A_k memorizes the disagreement $d = (s_i, s_j, r)$ in the list D. At this point, however, Untranslatability disagreements cannot be listed as they are not the result of a pairing relation between two concepts, but rather due to an absence of pairing relation.

7.4.3 Step 4: Search for Equivalences

At this point of Phase 2, both agents have a good representation of the relations between every concepts involved in the argumentation. However, before starting to resolve the disagreements, the agents need to fix two last issues, both related the the relations of equivalences present in the argumentation. First, the agents need to verify if one of the concepts that they recently added is a synonym of an already existing concept of their contrast set. If this is the case, the newest concept comes to replace the oldest one. Moreover, we mentioned in Section 6.5 that the agents need to ensure that no triadic relation comes as an infringement of the transitivity of our equivalence pairing relation. For these two reasons, the agents inspect closely the equivalence relations among their concepts before leaving Phase 2.

State 8: Send Own Local Internal Equivalences

- **Output Messages:** *Evaluation*
- **Next Possible State(s):** State 9

For each concept $C_i \in S_{K,k}$ and $C_j \in Add_K$ such that $C_i \neq C_j$, A_k computes the local pairing relation $C_i r_{Uk} C_j$ to check if:

- $C_i \equiv_k C_j$, or
- $C_i \dagger_k C_j$, or
- $C_i \ominus_k C_j$.

If this is the case, A_k might have to get rid of the old concept C_i, that is either redundant with C_j or even makes C_j indistinguishable as a concept (see indistinguishable disagreement in Section 4.5.3). In order to be sure, A_k will compute the overall pairing relation between C_i and C_j with the help of A_{-k}. This will be done similarly to how A_k and A_{-k} determined the overall pairing relations during Step 3. This means that A_k starts the process by sending the local r-triplet $r = r_l(C_i, C_j, U_k)$ to A_{-k} with a message *Evaluation*#9($id(C_i)$, $id(C_j)$, r).

State 9: Verify Other's Local Internal Relations

- **Input Messages:** *Evaluation*
- **Output Messages:** *Evaluation, Examples*
- **Next Possible State(s):** State 10

During this state, A_k will compute the loose r-triplets of each pairing relation that A_{-k} suspects to be problematic. In doing so, A_k helps A_{-k} to ensure that old concepts which are suspected to be either in a pairing relation of equivalence, one-sided or in no pairing relation with a new concept, are indeed in that relation and need to be removed. For each message *Evaluation*($id(C_i)$, $id(C_j)$, r) in its mailbox, A_k computes the loose r-triplet $r' = r_{ol}(C_i, C_j, U_O)$. Then, A_k sends a message *Evaluation*#10($id(C_i)$, $id(C_j)$, r') to A_{-k}.

For each unknown value $r'[x]$ from the r-triplet r', A_k sends the pairing partial set $U = U_k(x, C_i, C_j)$ to A_{-k} if $|U_k(x, C_i, C_j)| \leq |U_{-k}(x, C_i, C_j)|$ through a message *Examples*#10(U). With r and the local pairing partial sets, A_k will be able to tighten r and find one of the two necessary tight r-triplet that are needed to find the overall secured r-triplet.

State 10: Secure Own Overall Internal Relations

- **Input Messages:** *Evaluation, Examples*
- **Output Messages:** *Evaluation, Examples*
- **Next Possible State(s):** State 11

A_k starts its turn by adding all the examples received in its mailbox to its current contrast set and hypothesis contexts. Doing so, A_k will build a context that can be used to secure the loose r-triplets that it has also received from A_{-k}. For each message *Evaluation*($id(C_i)$,$id(C_j)$,r) that A_k has in its mailbox, A_k computes the tight r-triplet $r' = r_{ot}(r, U_k)$. This triplet r' is the first of the two necessary tight r-triplet that are needed to find the overall secured r-triplet according to the method described in Section 6.3.2. Then, A_k sends a message *Evaluation*#11($id(C_i)$, $id(C_j)$, r') to A_{-k} so A_{-k} can find the overall secured r-triplet once it has computed the second tight r-triplet.

Then, for each unknown value $r[x]$ from the loose r-triplet r, A_k sends the pairing partial set $U = U_k(x, C_i, C_j)$ to A_{-k} if $|U_k(x, C_i, C_j)| < |U_{-k}(x, C_i, C_j)|$ through a message *Examples*#11(U). Once again, A_k does so to allow A_{-k} to find the second tight r-triplet needed to compute the secured r-triplet that A_k needs.

State 11: Secure Other's Overall Internal Relations

- **Input Messages:** *Evaluation, Examples*
- **Output Messages:** *Remove*
- **Next Possible State(s):** State 12

In this state, the agent A_k has finally gather enough information to compute the pairing relation that A_{-k} wanted to verify in State 8. A_k will be able to directly tell A_{-k} which old concepts it should remove and which one it should keep. A_k starts its turn by adding all the examples that it has received in its mailbox to both of its current contexts $U_{K,k}$ and $U_{H,k}$. Then, for each message *Evaluation*($id(C_i)$,$id(C_j)$,r') that A_k has in its mailbox, A_k computes the tight r-triplet $r'' = r_{ot}(r, U_k)$, where r is the loose r-triplet $r_{ol}(C_i, C_j, U_O)$ computed by A_k during its last turn in state 9. With r' and r'', the agent A_k can compute the secured r-triplet $r^* = r_{os}(C_i, C_j, U_O)$ by using Conjecture 7.

With the binarized r-triplet $r_b(r^*)$, the agent A_k can check if:

- $C_i \equiv_O C_j$, or
- $C_i \dagger_O C_j$, or
- $C_i \ominus_O C_j$.

If this is the case, A_k sends a message *Remove*#13($id(C_j)$) to A_{-k}, in order to notify A_{-k} that the concept C_j is indeed compatible with the new concept C_i and therefore should be taken out of K_{-k}.

State 12: Send External Equivalences

- **Output Messages:** *Intransitive*
- **Next Possible State(s):** State 13

The agents are left with one task to finish in Step 4: verifying that the transitivity of the equivalence relation is kept in the argumentation, as explained in Section 6.5. During this state, the agent A_k intends to warn A_{-k} if A_{-k} has more than one concept that are equivalent to one concept of A_k. For each concept C_i in its contrast set, the agent A_k looks for the set of concepts $Int = \{C_j \in S_{K,-k} | \exists (C_i \equiv_O C_j) \in RT\}$. A_k sends a message *Intransitive*$\#13(\{id(C_j) | C_j \in Int\})$ to A_{-k} in order for A_{-k} to choose which of the concepts that are breaking the transitivity rule should be kept.

7.4.4 Step 5: Update Containers

In the previous steps of this phase, the agents have listed their pairing relations but have also identified and listed which concepts could not coexist within a same contrast set. During Step 5, the concepts that are either redundant with newer ones, or that break the transitivity rule of the equivalence pairing relation, are going to be removed by their agent. Once the contrast sets are finally compliant to Definition 49, the agents will be able to list the last type of disagreements that they were unable to identify during state 7: the untranslatable disagreements.

State 13: Update Contrast Set

- **Input Messages:** *Remove, Intransitive*
- **Output Messages:** *Remove*
- **Next Possible State(s):** State 14

For each message *Remove*$(id(C_i))$ in its mailbox, A_k deletes the concept C_i from its contrast set as A_{-k} identified this concept as redundant in Step 4. A_k notifies A_{-k} that it has deleted C_i by sending back a message *Remove*$\#14(id(C_i))$ to A_{-k}. Then, for each message *Intransitive*$(\{id_1, \ldots, id_n\})$ in its mailbox, the agent A_k search for the concept C_j from $Int = C(id_1, A_k), \ldots, C(id_n, A_k)$ which has the largest adjunct set in U_k. Then, the agent A_k removes all the concepts from $C(id_1, A_k), \ldots, C(id_n, A_k)$ except for C_j. Doing this, A_k makes sure that there are no more multiple concepts from its contrast set that are equivalent to one same concept from the other agent's contrast set. At the same time, selecting the concept with the more examples in its adjunct set limits the number of examples that will not be a part of a concept's extensional definition anymore, therefore maximizing the coverage of U_k by the new contrast set. A_k notifies A_{-k} that it has deleted C_j by sending back a message *Remove*$\#14(id(C_j))$ to A_{-k}. For each concept C removed from its contrast set, A_k removes any disagreement involving C from D.

State 14: Update Hypothesis

- **Input Messages:** *Remove*
- **Next Possible State(s):** State 15

While in state 13, A_{-k} has removed some concepts from its contrast set. A_k has to mirror this in its hypothesis, in order to keep H_k as close as possible from K_{-k}. For each message *Remove*(id_i) in its mailbox, the agent A_k removes the concept $C(id_i, A_{-k})$ from its hypothesis. For each concept C removed from its hypothesis, A_k removes any disagreement involving C from D.

Now, the agents have hypotheses that match each others contrast sets, with each concept of a contrast set equivalent to a maximum of one other concept from another contrast set. All the pairing relations are are listed, along with the disagreements to resolve. The agents are ready to move to the next phase.

7.5 Phase 3: Resolve Disagreements

7.5.1 Step 6: Choose Disagreement

Step 6 is a one-state step that allows the agents to pick a disagreement to resolve during the rest of Phase 3. Before selecting the disagreement, the agents will sort their set of unresolved disagreements in order to optimize the disagreements resolution. If the agents do not have any disagreement left to resolve, they can end their argumentation and eventually move to the optional Phase 4 to clarify their vocabularies.

State 15: Choose Disagreement

- **Output Messages:** *Debate*
- **Next Possible State(s):** State 16, 17, 22, 23 or 24

Before picking a disagreement to resolve, A_k needs to make sure of two things:

- to list the untranslatable disagreements that A_k could not list during Step 3, and
- to check that A_{-k} has not already gone through state 15 during this phase, and has not already picked a disagreement to resolve.

First, A_k looks for each concept $C_i \in S_{K,k}$ that is not in an equivalence relation with a concept from $S_{K,-k}$. Recall that A_k kept track of the overall relations between all the concepts in its list RT. Then, A_k adds the disagreement $d = (s(C_i), \bullet, C_i \not\equiv_O \bullet)$ to the list of disagreements D. After adding all the untranslatable disagreements to D, the agent A_k has listed all the overall disagreements that are currently present in the argumentation. A_k is now ready to either pick a disagreement to resolve, or to accept a new proposition for a disagreement to discuss from A_{-k}.

If A_k has a *Debate*(d) message in its mailbox, it sets up its current disagreement d_c as d. If A_k has not received a *Debate* message, it is the first agent to enter state 15: A_k picks a disagreement d from D, sets it as its current contrast set d_c and sends a message *Debate*(d) to A_{-k}. Disagreements are picked in the order presented in Section 5.5. A_k sets its next state as:

- 16 if d_c is a self-disagreement,
- 17 if d_c is a semantic or untranslatable disagreement,
- 25 if d_c is an indistinguishable disagreement,
- 26 if d_c is a lexical disagreement and
- 27 if there are no disagreement in D to address anymore.

7.5.2 Step 7: Build Extensional Definition

The agents enter Step 7 when the creation of a new concept is necessary to the resolution of the ongoing disagreement. The creation of the desired extensional definition for the new concept is the first step to every concept creation. While the standard procedure is to create one concept at a time, the agents that are resolving a self-disagreements will need to create two concepts at the same time. In this case, two extensional definitions are created at the same time.

State 16: Fix Boundaries

- **Output Messages:** *Size*
- **Next Possible State(s):** State 19

The procedure to resolve a self-disagreement is detailed in Section 5.5. Since resolving a self-disagreement $d_c = C_i \otimes_O C_j$ implies to replace the two concepts C_i and C_j involved in the disagreement by two new concepts C_i' and C_j' through argumentation. The first step in the creation of a new concept through argumentation is the creation of the set of positive and negative examples that will be used to generate the new intensional definition by inductive learning. The agent A_k will therefore create two sets of positive examples U_i^+ and U_j^+, and two sets of negative examples U_i^- and U_j^-. Once these four sets have been created, the agent A_k will decide if it takes the leadership of the concept creation as explained in Section 5.1.3. Then, A_k sends two messages $Size\#19(|U_i^+|)$ and $Size\#19(|U_j^+|)$ to A_{-k}, in order to help A_{-k} to determine the maximal number of positive and negative examples that it should allow during the creation of the intensional definition.

State 17: Build Extensional Definition

- **Output Messages:** *Size*
- **Next Possible State(s):** State 18

The procedure to resolve semantic and untranslatable disagreements is detailed in Section 5.5. In the case of an overlap disagreement the agent A_k creates a set of positive examples U_i^+ that corresponds to the overlap between the two involved concepts. In the case of a hypo/hypernymy disagreement, A_k creates a set of positive examples U_i^+ that corresponds to the co-hyponym of the two concept's hyponym. In the case of an untranslatable disagreement, A_k creates a set of positive examples U_i^+ that corresponds to the untranslatable concept's adjunct set. Then, A_k sends a message $Size\#19(|U_i^+|)$ to A_{-k}, in order to help A_{-k} to determine the maximal number of positive and negative examples that it should allow during the creation of the intensional definition.

State 18: Determine Leadership

- **Next Possible State(s):** State 19

The current disagreement d_c involves a pairing relation between two concepts, and the agents going through state 18 means that this relation will push them to create one or two new concept(s). Before creating the new concept, the agents need to decide which one of them will, for each concept to create, lead the argumentation. This choice is detailed in Section 5.1.3. According to the method described in Section 5.5.5, the agent A_k has access to all the resources that it needs to know if it should be in charge of the argumentation of not. If A_k observes that it should be in charge of the argumentation, it notifies A_{-k} by sending a message $Seize\#19()$.

7.5.3 Step 8: Build Intensional Definition

- **Next Possible State(s):** State 17

Once the extensional definition(s) of the concept(s) that will be created have been defined, the agents will use them to create the intensional definition of the new concept(s). The process has already been detailed in Section 5.1.3, and will not be presented again in this Chapter.

State 19: Build Intensional Definition

- **Next Possible State(s):** State 20, 22.

Now that the agents each built positive and negative sets of examples, chose a leader for the argumentation and determined which were the maximum acceptable false positive and negatives during the argumentation, they can go through the creation of the intensional definition(s) of the new concept(s), as explained in Section 5.2. If the creation of the intensional definition(s) is successful, A_k continues to state 20 in order to create the remaining elements of their new concept(s). If the agents could not generate a satisfying intensional definition through inductive learning, A_k moves to state 24 in order to remove the concepts involved in the disagreement.

7.5.4 Step 9: Build New Concept

Once the agents have the extensional definition(s) and the intensional definition(s) of the new concept(s), they can finish to create it by adding it a sign and an identifier. Once all the elements of the new concept(s) has been assembled, the agents can move to the beginning of Phase 2 in order to add them to their contrast sets and evaluate their impact on the argumentation.

State 20: Build Sign and Identifier

- **Output Messages:** *Baptize*
- **Next Possible State(s):** State 21

If the agent A_k has not received a message *Seize*() in its mailbox, it knows that it is the first agent to go through state 20 during this phase. Therefore, A_k creates a new sign s_i , as explained in Section 5.1.3, and a new identifier id_i for each concept C_i that is being created. A_k shares each created sign and identifier with A_{-k} through a message *Baptize*#21(s_i, id_i). Then, A_k sends a message *Seize*#20() to A_{-k} in order to let him know that it has created the new signs. In the case where multiple concepts are being created, an element is added to each *Baptize* message in order to determine which sign and identifier should be assigned to which new concept.

State 21: Build Concept

- **Input Messages:** *Baptize*
- **Next Possible State(s):** State 2

For each concept C_i that is created, the agent A_k gets:

- the extensional definition $E_i = U_i^+$ created in state 16 or 17,

- the intensional definition I_i from the accepted root-argument found in state 19
- the sign s_i and id_i either determined in state 20 or received through a *Baptize*() message,

and creates the new concept $C_i = \langle s_i, I_i, E_i \rangle$ with the identifier id_i. Depending to the type of the current disagreement d_c, the new concept C_i will go to a different container, according to the protocol specified in Section 5.5. In the case of a self-disagreement, A_k will add C_i to K_k if d_c involved concepts from its contrast set, and to H_k otherwise. If d_c was an overlap disagreement, A_k adds C_i to both K_k and H_k. If d_c was a hypo/hypernymy disagreement, A_k adds C_i and a copy of its co-hyponym to K_k if A_k if the hypernym of the disagreement belonged to A_k, or adds them to H_k otherwise. If d_c was an untranslatable disagreement, A_k adds C_i to K_k if A_k was the agent missing a concept in its contrast set, and to H_k otherwise. Moreover, according to Section 5.5, if d_c was a hypo/hypernymy disagreement A_k needs to remove the hypernym C_H from whichever container it is in. The disagreements that involved C_H are therefore removed from D, as they are now resolved. Each concept added to K_k or H_k is added respectively to Add_K or Add_H. Finally, A_k removes the disagreement d_c from D if it was not already done.

7.5.5 Step 10: Delete Concept

Sometimes, the agents cannot resolve a disagreement – either because they are blind to this disagreement or because their inductive learning performances do not allow them to achieve the generalizations they aimed for – and are forced to delete concepts from their contrast sets without replacing them. This deletion will leave a blank in the agents' contrast sets, but allows them to move toward a synchronic agreement at the cost of their context coverage.

State 22: Delete Involved Concept

- **Input Messages:** *Remove*
- **Output Messages:** *Remove*
- **Next Possible State(s):** State 15

If the creation of the new concept(s) through argumentation has not been successful, the only option left to the agents is to delete one of the concepts that were involved in $d_c = C_1 r_O C_2$, in order to resolve the disagreement at the cost of a loss of context coverage for the current contrast sets. The agent A_k checks whether or not it received a message $Remove(id(C_i))$ in its mailbox. If it did, A_k removes the concept C_i from whichever container it is in. If A_k did not receive a message, it selects the concept C_i from C_1 and C_2 that has the smallest adjunct set, or randomly select one if their adjunct sets have the same size. Then, A_k sends a message $Remove\#(C_i)$ to A_{-k} in order to make A_{-k} delete the same concept. At the end of its turn, the agent A_k removes the disagreements involving C_i from D as they are now resolved.

7.5.6 Step 11: Change Signs

Some disagreements require the creation of new concepts, but some others can be resolved simply by changing the signs of some concepts. Changing the signs of the concepts is often done when the agents have resolved all the disagreements that requested a concept creation, as changing the sign of a concept before the same concept being deleted is a waste of resources.

State 23: Update Sign(s)

- **Input Messages:** *Replace*
- **Output Messages:** *Replace*
- **Next Possible State(s):** State 15

If $d_c = C_i r_O C_j$ is a lexical disagreements, the agent will have to substitute the signs of some of their concepts, according to the protocol described in Section 5.5. If d_c is a synonymy and A_k received a message *Replace*(s', C_i), A_k changes the signs of both C_i and C_j by s'. If d_c is a synonymy but A_k did not receive such message, A_k creates a new sign s', sends a message *Replace*$\#23(s', id(C_i))$ to A_{-k}, and then changes the sign of both C_i and C_j to s'. If d_c is a homonymy and A_k received two messages *Replace*(s_1, C_i) and *Replace*(s_2, C_i), A_k changes the signs of C_i to s_1 and the sign of C_j to s_2. If d_c is a homonymy but A_k did not receive such messages, A_k creates two new signs s_1 and s_2, sends two messages *Replace*$\#23(s_1, id(C_i))$ and *Replace*$\#23(s_2, id(C_j))$ to A_{-k}, and then changes the sign of C_i to s_1 and C_j to s_2. Then, A_k deletes the disagreement d_c from R.

7.6 Phase 4 (Optional): Update Vocabulary

Phase 4 is optional, meaning that going through it or not will not impact the performances of the multi-agent system during a test – unless, of course, for the time of completion that will be slightly higher. However, Phase 4 has a cosmetic aspects in the sense that replaces the automatically generated signs created during the different states of the argumentation, by the original signs of the initial contrast sets. This allows the final contrast sets to have signs similar to *Chair* or *Bird* instead of *sign-0* or *label-1*.

7.6.1 Step 12: Update Vocabulary

During the unique step of Phase 4, the agents will try to replace the sign of each of their final concepts by the sign of their initial concept that was the closest to it, without assigning the same sign twice to a same final concept. The agents will send to each other, for every concept C_i in their initial contrast set and each concept C_j in their final contrast set, how confident they are that the concept C_j is similar enough to C_i for substituting $s(C_j)$ with $s(C_i)$.

State 24: Vote for Signs

- **Output Messages:** *Vote*
- **Next Possible State(s):** state 25

A_k sends a message *Vote*$\#25(s_i, s_j, n)$ to A_{-k} for every sign s_i currently used in K_k and every sign s_j that was used in the initial contrast set $K_{i,k}$. The number n is the number of examples from the concept $C_i \in K_k$ that were initially in the concept $C_j \in K_{i,k}$.

State 25: Elect Signs

- **Input Messages:** *Vote*

A_k creates the set of signs used in the old contrast sets S_{old} by adding to S_{old}, for each message $vote(s_j, s_i, n')$ that A_k has received and each message $Vote(s_i, s_j, n)$ that A_k sent in the last state, the sign s_j. A_k also creates a set of signs from the new contrast sets S_{new} and adds to S_{new}, for each message sent in the last state and each message received, the sign s_i.

Then, A_k will choose which sign $s_i \in S_{new}$ should be replaced by which sign $s_j \in S_{old}$. Starting with a random sign $s_j \in S_{old}$, A_k looks for each sign $s_i \in S_{new}$, and calculates a numerical value v_{s_i} equals to:

- $n + n'$ if $Vote(s_i, s_j, n)$ has been sent and if $vote(s_j, s_i, n')$ has been received.
- n if no message $Vote(s_j, s_i, n')$ has been received.
- n' if no message $Vote(s_i, s_j, n)$ has been sent.

Then, A_k takes the sign s_{max} associated to the highest of these numerical values $v_{s_{max}}$ and then replaces s_{max} by s_j in all its concepts. Then, A_k removes s_j from S_{old} and s_{max} from S_{new}.

A_k picks another random sign s_k from S_{old} and repeats this procedure until either S_{old} or S_{new} becomes empty.

7.7 Conclusion

We presented the systematic strategy, a strategy of argumentation in which the agents verify each other's meaning before entering the naming game. In this approach, the agents systematically exchange their concepts upon meeting and identify at once all the disagreements present between their contrast sets. Once identified, all the examples are resolved by the agents in order for them to attain a mutual intelligibility on the full extent of their overall context. The proposed model is an error-tolerant model that will not consider small overlaps between concepts as disagreements. This model is separated in four main phases that characterize the progression of the argumentation between our agents, from their meeting to reaching mutual intelligibility. The first phase is the initial transfer of intensional definitions made by the agents upon meaning. The second phase is the identification of disagreements trough the exchange of r-triplets and inference of overall pairing relations. The third phase is the resolution of each type of disagreement. The fourth optional phase is the creation of an intelligible vocabulary.

Chapter 8

Lazy Strategy to Mutual Intelligibility

8.1 Introduction

The lazy strategy is our second approach to reach mutual ineligibility. As the systematic strategy, it is based on the model presented in Chapters 4 and 5. The lazy strategy consists of playing a naming game where examples are presented to the agents, until the two agents name an example differently. The agents then identify the disagreement d responsible for this difference, and identify d's set of connected disagreements D. The concepts involved in this set of disagreements are extracted from the contrast sets of our agents K_1 and K_2, and isolated in a new pair of containers Q_1 and Q_2. Using phases similar to Phase 2, Phase 3 and Phase 4 in the systematic strategy, the agents address the disagreements of D one by one until D becomes empty. The concepts of Q_1 and Q_2, which are now causing no disagreements, are reinserted in K_1 and K_2 so the agents reach a *partial* mutual intelligibility concerning the original disagreement. After the argumentation, the agents can continue to play the naming game until another disagreement is found. The lazy strategy is described below takes into consideration a degree of error τ.

8.2 Structure of Argumentation in the Lazy Strategy

While the systematic strategy described in Chapter 7 that was characterized by its linear structure that was bearing only one loop, the lazy strategy is organized as a loop that encompasses several other loops. The main loop is a continuous naming game, starting at the presentation of a new example and exiting when no disagreement between the agents can be suspected. The loop is divided in five main Phases: *Monitoring Loop*, *Connected Set of Disagreement Retrieval*, *Evaluation*, *Resolve Disagreements*, the optional phase *Update Vocabulary*, and *Knowledge Integration*. The entry point of the lazy strategy is the Monitoring Loop, within which the agents cycle until they receive a new example that sheds light upon a disagreement. Since the naming game is open-ended, in the sense that there can always be a new example presented to the agents for them to name, there is no predefined end to the lazy strategy —except the lack of new examples.

Each phase is divided in main steps, that identify clear goals to walk toward the objective of each phase. For instance, *Monitoring Loop* is divided in three main Steps that each deal with a particular aspect of the naming of a new example in the context of naming game: the step *Naming a New Example* watches for new a example to name, the step *Name New Example* associates names to a new example, which the agents share them with one another, and the step *Compare Naming* identifies a success (agreement) or a failure (disagreement) within the naming game.

The phases and steps of the lazy approach are represented in Figure 8.1. Each step is divided in states, as before. States are include a decision procedure followed during the turn of an agent. Each state starts with receiving a token and ends by passing this token to the other agent. The last actions defined in a state are some of the following: to send a messages to the other agent, to chose the next state in which the agent should be next turn, to remove messages from its mailbox, and to pass the token. The phases, steps and states of the lazy strategy are described in the following sections of this chapter. The lazy strategy is described from the point of view of an agent A_k (the one holding the token) that is arguing with the other agent A_{-k} —where $k \in \{1, 2\}$.

8.3 Phase 1.a: Monitoring Loop

During this phase, the agents wait for a new example to be presented to them. Once the example is presented, the agents are naming it and exchanged the sign they used so each of them can judge of the success of this round of the naming game. At the end of the monitoring loop, the agents move toward an argumentation of the signs they used to name the example were different or go back to wait for a new example otherwise.

8.3.1 Step 1: Naming a New Example

The agents entry point in the argumentation is a stand-by state where they wait for an example to be presented to them. They exit this state by receiving an example that initiates the naming game. When the experimenter wants to start a round of naming game, it presents an example e to both agents A_1 and A_2 through a message *Examples*$\#1(e)$ sent in both agents' mailboxes.

State 1: Stand-by

- **Input Messages:** *Examples*
- **Next Possible State(s):** State 1, 2

If the agent A_k has not received a new example to name, A_k stays in the same state next turn waiting for an example to play the naming game. Therefore, if A_k's mailbox is empty, A_k sets its next state as State #1 and passes the token to A_{-k}. Otherwise, if A_k has received a message *Examples*(e) in its mailbox, A_k can start the naming game. A_k sets its next step as state 2 to continue the naming game.

8.3.2 Step 2: Name a New Example

Once the naming game has started, the first thing that the agents have to do is to name the example that has been presented to them. The agents name this example with the signs of the concepts that subsume it and share these signs with each others.

State 2: Name

- **Output Messages:** *Self-Check, Name*
- **Next Possible State(s):** State 3

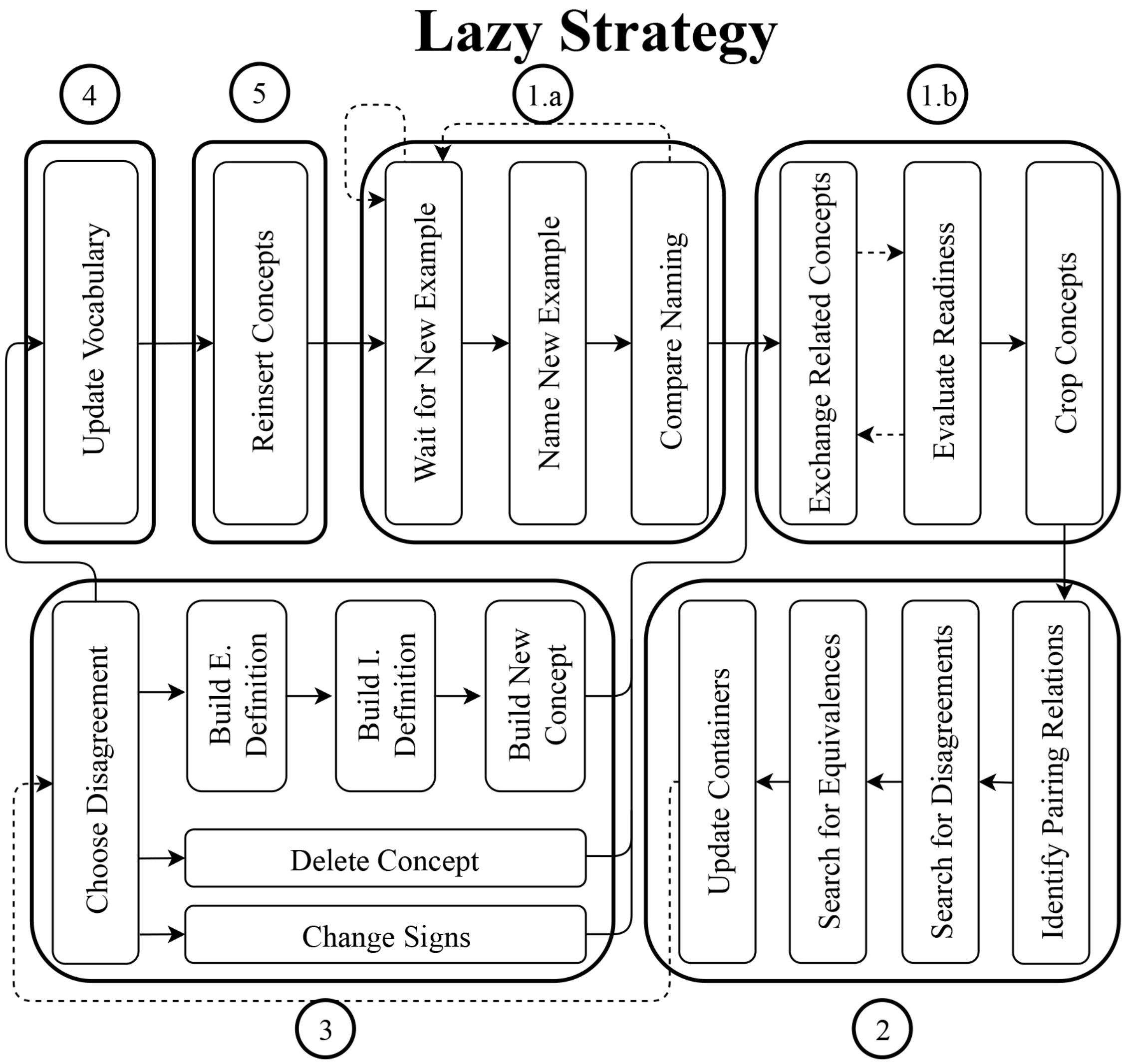


Figure 8.1: Diagram of the five main phases of the lazy strategy for argumentation (bold rectangles), with their respective steps (thin rectangles). Each step is linked to its possible next steps by an arrow. The dashed-line arrow symbolizes the loops of our argumentation strategy in Phase 1.a, Phase 1.b and between the phases 2 and 3.

During the last state, A_k defined the example e_c as the example to be named during this current round of the naming game. A_k searches in its current contrast set $K_c = (S_c, U_c)$ for the set of concepts $S_{ec} = \{C \in S_c | C \sqsubseteq e_c\}$ that subsume the example e_c. Then, A_k sends a message *Name#2*$(s(C_i), e_c)$ to A_{-k} for each concept C_i in S_e. Additionally, A_k sends a message *Self-Check#7*() to A_{-k}, in order to remind A_{-k} to look for the pairing relations between its own concepts, as the agents are now entering a new round of the naming game. As in the systematic strategy, the agents need to search for the pairing relations of all their concepts the first time they start to investigate the disagreements between two sets of concepts. Later on, the agents will just have to update some targeted relations when some concepts change.

8.3.3 Step 3: Compare Naming

Once both agents have named the example e_c, they can compare the signs that have been used to name it. In the naming game, only one sign is acceptable to name e_c. For this reason, if even one agent uses multiple signs to name e_c, the agents are going to move forward an argumentation.

State 3: Compare

- **Input Messages:** *Name*
- **Output Messages:** *Assert*
- **Next Possible State(s):** State 1, 4

The naming game is considered as a success if both A_1 and A_2 used the same identical sign to name the example e_c. In order to evaluate the success of this round of the naming game, the agent A_k creates a set of signs $S = \{s_i, \ldots, s_m\}$ with the signs that A_k sent to A_{-k} during the last state 2. Then, A_k searches its mailbox for the messages *Name*$(s_n,\ e_c)$, ..., *Name*$(s_p,\ e_c)$ sent by A_{-k} and creates the set of signs $S' = \{s_n, \ldots, s_p\}$.

If both sets of signs contain only one sign such that $S = \{s\}$ and $S' = \{s\}$, then the two agents have succeeded in the naming game and can stop the argumentation. The agent A_k sets its next state to state 1 and passes the token to A_{-k}. Otherwise, the agents know that there are at least two concepts with different signs that are subsuming one same example. While one example subsumed by both concepts is not enough to cause a disagreement if $\tau > 1$, the agents will still investigate the concepts for disagreements.

The first thing that A_k does is to check if disagreements caused by the same concepts have not been looked for or resolved before. A_k keeps a list of concepts that have been investigated for disagreements in two lists: a list $Checked_k$ for its concepts and a list $Checked_{-k}$ for A_{-k}'s concepts. Before starting an argumentation, the agent A_k verifies that it has not already been though that process by checking, for each sign $s \in S$, if there is a concept $C \in S_{K,k}$ such that $s(C) = s$ and $C \in Checked_k$. A_k also verifies the concepts of the other agent by checking, for each sign $s' \in S'$, if there is a concept $C' \in S_{K,-k}$ such that $s(C') = s'$ and $C' \in Checked_{-k}$. If all the concepts of both agents have already been investigated for disagreements, the differences in signs during the naming game are considered as an acceptable error under τ and the agent A_k does not pursue the argumentation. The agent A_k sets its next state to state 1 and passes the token to A_{-k}.

If at least some concepts have not been investigated, the agents look for disagreements caused by them. In order to find disagreements, the agents need to be aware of the overall pairing relations between their concepts and in order to be aware of the overall pairing relations between their concepts, the agents need the intensional definitions of their concepts. For this reason, the agents start to exchange the intensional definitions of the concepts involved in the disagreement. For each sign s in $S = \{s_i, \ldots, s_m\}$, the agent A_k sends an assert message to A_{-k}. If the set

$Checked_k$ contains a concept C such that $s(C) = s$, the concept C has already been investigated for disagreements meaning that A_k has already sent the intensional definition $I(C)$ to A_{-k}. In order to limit the number of generalizations exchanged by the agents, the agent A_{-k} send a message $Assert\#4(s(C), id(C))$ to A_{-k}. Otherwise, if there is no such concept C in $Checked_k$, A_k has to send the intensional definition of of its concept $C \in S_{K,k}$ that has a sign $s(C) = s$ to A_{-k}. Therefore, A_k sends a message $Assert\#4(s(C), id(C), I(C))$ to A_{-k} and ends its turn. Since it will now be investigated for disagreements, the concept C is added to $Checked_k$ before A_k ends its turn.

8.4 Phase 1.b: Connected Set of Disagreements

In the lazy strategy, an argumentation targets a specific set of connected disagreements D. In order to argue on the disagreements from D with our model, the agents need to both have access to the intensional definitions of every concept that causes a disagreement in D. Unlike the systematic strategy, where the agents would have already exchanged all their intensional definitions with each other, the agents have no a priori knowledge on each other's contrast set in the lazy strategy. The agents do not need to exchange all their intensional definitions, as their argumentation is concerned only with the concepts from D. During this phase, the agents look for concepts that belong to D, exchange the intensional definitions of these concepts, and update their hypotheses.

Moreover, the agents keep a clear separation between the concepts that are not yet suspected to be in a disagreement and the concepts that are part of the argued connected set of disagreements. In order to keep that clear separation, the agents use new contrast sets called the *temporary* contrast sets, noted K_t, where they put the concepts involved in the set of connected disagreement that is currently argued. The agent A_1 has a temporary contrast set K_{t1} and the agent A_2 a temporary contrast set K_{t2}. Since the agents are going to use these contrast sets as the base container for their argumentation, they also need corresponding hypotheses. The agent A_1 has a temporary hypothesis H_{t1} that mirrors the concepts of K_{t2}, while the agents A_2 has a temporary hypothesis H_{t2} that mirrors the concepts of K_{t1}.

8.4.1 Step 4: Exchanging Related Concepts

The agents maintain a list of the concepts involved in the set of connected disagreement that is currently argued. During this step, the agents update this list, in an iterative process where each agent looks for concepts of its contrast set that are not in the list but that cause *local* disagreements with concepts from the list. Indeed, two concepts can have a local pairing relations does not cause a disagreement while their overall pairing does. This means that if the agents were to look for all the concepts causing overall disagreements, they would have to exchange all their intensional definitions with each other. For this reason, the agents cannot be sure that they have listed all the disagreements from the connected set that they target in the lazy strategy. The first time that the agents arrive in that step after step 3, the list is empty. The agents seed the list with the concepts that have been suspected of causing a disagreement in step 3, the intensional definitions of which have been exchanged at the end of state 3.

State 4: Concepts Listing and Sharing

- **Input Messages:** *Assert*
- **Output Messages:** *Assert,Seize*
- **Next Possible State(s):** State 5

The agent A_k creates a local copy of the concept C for each message $Assert(s(C), id(C), I(C))$ that A_k has received, using the method described in 5.1.3. When a message $Assert(s(C), id(C))$ has been received, the agent A_k has already a local copy of the concept C in its list $Checked_k$, and can therefore retrieve $I(C)$ using the concept C's identifier, $id(C)$.

A_k then adds the concept C to its list Add_H, as C will join the argumentation. Once every message $Assert$ has been read, the agent A_k has a list of concepts L that A_{-k} has requested to add to the argumentation. If the last state of A_k was State 3, these concepts are the two concepts that were suspected to cause a disagreement during Phase 1.a and that now seed the search for the rest of their connected set of disagreements.

For each pair of concepts C, C' such that $C \in L$ and $C' \in K_{c,k}$, the agent A_k evaluates if C and C' are causing a local disagreement in U_k. If C' causes a disagreement, the agent A_k sends an $Assert$ message to A_{-k} in order to add C' to the argumentation. In order to limit the number of augmentations exchanged, A_k send a message $Assert\#4(s(C'), id(C'))$ if C' is already in $Checked_k$, which means that A_{-k} had already received the intensional definition $I(C')$. Otherwise, the agent A_k sends a message $Assert\#4(s(C'), id(C'), I(C'))$ to A_{-k}. The concept C' joins the argumentation and is therefore added to the list Add_K. If no concept of $K_{c,k}$ was causing disagreements a concept of L, the agent A_k advises A_{-k} that it does not wish to add more concepts to the argumentation by sending a message $Seize\#5()$ to A_{-k}.

8.4.2 Step 5: Evaluate Readiness

Since the agents list the concepts to add to the list of concepts involved in the connected set of disagreements D based on their local pairing relations, one agent can list a concept that is ignored by the other. Because of this, each agent needs to re-evaluate which concepts should be added to the list of concepts involved in D's disagreements every time that the other agent has added new concepts to the list. For this reason, the argumentation can continue only once both agents consider that there are no more concepts to add to the list. During this step, the agents check with each other whether the argumentation should go back to step 4 or continue to the next step.

State 5: Conciliation on Pursuing Argumentation

- **Input Messages:** *Assert, Seize*
- **Next Possible State(s):** State 4, State 6

If A_k has not received any *Seize* message in its mailbox, A_k is informed that the agent A_{-k} does not wish to add more concepts to the list of concepts causing disagreements and has not send additional *Assert* messages that A_k needs to examine in State 4. If A_k did not find any new concepts to add to the argumentation during the previous step 4, A_k is also ready to continue the argumentation. A_k sets its next state to State 6 and ends its turn. However, if either A_k or A_{-k} had added new concepts to the argumentation during the previous step 4, the agents need to go back to step 4 in order to evaluate whether or not new concepts need to be added to the argumentation. The agent A_k sets its next state to State 4 and ends its turn.

8.4.3 Step 6: Crop Concepts

Once the concepts involved in the connected set of disagreement D are listed, the agents move all of them from their current contrast sets and current hypotheses to their temporary containers. This isolates the disagreements in separate containers and help to evaluate whether or not the expected partial mutual intelligibility has been reached. During this step, the concepts that are suspected to cause a disagreement are listed for transfer. Once the concepts have been listed for transfer, the agents can move to the next step to effectively transfer their concepts.

State 6: Concept Transfer to Temporary Containers

- **Next Possible State(s):** State 7

The agent A_k moves the concepts that have been listed as suspected of causing disagreements, from the current containers to the temporary ones. A_k removes each concept C from the contrast set $K_{c,k}$ and adds C to a list $transfer_K$ to remember that C has been removed from $K_{c,k}$ and should be transferred to $K_{t,k}$. Then, A_k removes each concept C' from the hypothesis $H_{c,k}$ and adds C to a list $transfer_H$, to remember that C' has been removed from $H_{c,k}$.

8.5 Phase 2, Phase 3, and Phase 4

For the lazy strategy Phases 2, 3 and 4 are the same as Phases 2, 3 and 4 in the systematic strategy except in four aspects. First, since the number of steps and states now Phase 1.a and 1.b is larger than in Phase 1 in the systematic strategy. Therefore, the step 7 of the lazy strategy corresponds to the step 2 of the systematic strategy, and state 7 of the lazy strategy corresponds to the state 3 of the systematic strategy. This offset in the numbering of the states and steps is maintained during all of Phases 2, 3 and 4. Tables 8.1, 8.2 and 8.3 summarize the offset and recalls the different steps and state of the Phases 2, 3 and 4.

The second difference with the systematic strategy regards the containers involved during the argumentation. In the systematic strategy, the agents argue about the concepts of their current contrast sets (K_1 and K_2), that are mirrored by the hypotheses (H_2 and H_1). In the lazy strategy, the argumentation does not take place with the concepts of the current containers but is restricted ot the concepts of the temporary containers comprising the two contrast sets $K_{t,1}$ and $K_{t,2}$ and the two hypotheses $H_{t,1}$ and $H_{t,2}$. Therefore Phases 2, 3 and 4 are now like those of the systematic strategy where current containers are substituted by temporary containers. However, the context of these containers are shared: for any agent A_k, the containers $K_{c,k}$, $K_{t,k}$, $H_{c,k}$ and $H_{t,k}$ all share the same context U_k. Any example added to the context of $K_{t,k}$ or $H_{t,k}$ during the phases 2, 3 or 4 is also added to the context of $K_{c,k}$ and $H_{c,k}$.

The third difference with the systematic strategy regards the end of phase 3. Instead of looping directly back to phase 2 after the states 23, 26 and 27 (previously numbered 19, 22 and 23), the agents are looping back to the state 7, at the beginning of step 7. This forces the agents to look for new disagreements that should be integrated to the connected set of disagreements currently under investigation every time that a modification has been made to the temporary contrast sets.

The last difference with the systematic strategy regards the signs that are used during the optional Phase 4. Instead of using the signs of their initial vocabulary, as it was the case in the systematic strategy, the agents only use the signs from the concepts that have been transferred from their current contrast sets to their temporary contrast sets. The agents do so in order to avoid using a sign that is still used by a concept of the current contrast set, in a concept of their temporary contrast set.

8.6 Phase 5: Knowledge Integration

In Phase 1.b, the agents have progressively isolated a set of concepts from their current contrast sets in temporary containers. The concepts from this set were causing synchronic disagreements

Lazy Strategy	Systematic Strategy	Note
Step 7	**Step 2**	**Identifying the pairing relations of the concepts added to the argumentation.**
state 7	state 3	Computing and sending the local R-Triplets.
state 8	state 4	Computing and sending the loose R-Triplets.
state 9	state 5	Computing and sending the tight R-Triplets.
state 10	state 6	Computing the secured R-Triplets.
Step 8	**Step 3**	**Search for Disagreements**.
state 11	state 7	Find disagreements.
Step 9	**Step 4**	**Search for Equivalences.**
state 12	state 8	Look for locally equivalent concepts in own contrast set.
state 13	state 9	Look for locally equivalent concepts in other agent's contrast set.
state 14	state 10	Secure the R-Triplets of pairing relations suspected to be equivalences in own contrast set.
state 15	state 11	Secure the R-Triplets of pairing relations suspected to be equivalences in other agent's contrast set.
state 16	state 12	Verify the transitivity of the equivalence pairing relation.
Step 10	**Step 5**	**Update Containers.**
state 17	state 13	Removing duplicated concepts from contrast set.
state 18	state 14	Removing duplicated concepts from hypotheses.

Table 8.1: Correspondences between the steps and states of the lazy and systematic strategies during Phase 2. The different steps are separated by a double line.

Lazy Strategy	**Systematic Strategy**	**Note**
Step 11	**Step 6**	**Choose Disagreement.**
state 19	state 15	Select disagreement to be resolved in the rest of the phase.
Step 12	**Step 7**	Build Extensional Definition.
state 20	state 16	Distribute the examples from a self-disagreement's overlap.
state 21	state 17	Isolate the examples that the new concept is supposed to subsume.
state 22	state 18	Determine which agent will lead the creation of the new concept.
Step 13	**Step 8**	Build Intensional Definition.
state 23	state 19	Create the concept's new intensional definition through argumentation.
Step 14	**Step 9**	**Build New Concept.**
state 24	state 20	Create the sign and the identifier of the new concept.
state 25	state 21	Create a new concept with the semiotic elements generated in the previous states, in order to resolve a self-disagreement, an overlap of a hypo/hypernymy disagreement. Remove the hypernym of this concept if there is one.
Step 15	**Step 10**	**Delete Concept.**
state 26	state 22	Delete one of the concepts involved in a disagreement in order to resolve that disagreement.
Step 16	**Step 11**	**Change Signs.**
state 27	state 23	Replace the signs of some concepts by new signs in order to resolve homonymy and synonymy disagreement.

Table 8.2: Correspondences between the steps and states of the lazy and systematic strategies during Phase 3. The different steps are separated by a double line.

Lazy Strategy	**Systematic Strategy**	**Note**
Step 17	**Step 12**	**Update Vocabulary.**
state 28	state 24	Vote to replace the signs of the agents current vocabulary with signs from their initial contrast sets vocabulary.
state 29	state 25	Update the vocabulary of the agents with the vocabulary of their initial contrast sets according to the votes.

Table 8.3: Correspondences between the steps and states of the lazy and systematic strategies during Phase 4. The different steps are separated by a double line.

that have later been resolved during Phases 2 and 3. Their resolution resulted in another set of concepts free of disagreements, which are the current concepts of the temporary containers. and that the agents reinserts in their current contrast sets in order to replace the concepts that had been removed during Phase 1.b.

8.6.1 Step 18: Reinsert Concepts

After creating new concepts or substituting signs during Phase 3, and after updating their vocabulary during Phase 4, the agents are always passing through Phase 2 where they verify that there are no disagreements between the concepts of the temporary contrast sets and the concepts of the current contrast sets. Therefore, the agents can just insert the concepts from their temporary contrast sets in their current contrast sets without creating any new disagreements. Once the concepts from the temporary contrast sets have been transferred back to the current contrast sets, the agents have reach partial mutual intelligibility over the part of their context that is subsumed by concepts from the temporary contrast sets. This part of mutual intelligibility adds itself to any previous partial mutual intelligibility that the agents had already reached.

State 30: Concept Transfer to Current Containers

- **Next Possible State(s):** State 1

The agent A_k updates the list of concepts that have been investigated for disagreements. A_k starts by removing the concepts of $\mathit{transfer}_K$ from $\mathit{checked}_K$ and the concepts of $\mathit{transfer}_H$ from $\mathit{checked}_H$. Then, A_k replaces the removed concepts by the concepts of the temporary containers. A_k places the concepts of $K_{t,k}$ in $K_{c,k}$ and the concepts of $H_{t,k}$ in $H_{c,k}$. Now that the partial mutual intelligibility has been reached over the portion of the overall context that was concerned by the investigated connected sets of disagreements, the agent A_k prepares to potential future augmentations. A_k empties the lists $\mathit{transfer}_K$ and $\mathit{transfer}_H$, and Add_K and Add_H. Then, A_k removes every concepts from $K_{t,k}$ and $H_{t,k}$. The argumentation over the meaning is over and the agents go back to state 1, waiting to test another example for possible disagreements.

8.7 Chapter Conclusion

We have presented the lazy strategy, a strategy that does not assume the systematic exchange of intentional definitions between the agents before the play of the naming game. During the naming game, the agents will be able to identify communication failures and spot disagreements. In this approach, the resolution of disagreement is operated when disagreements are detect. The proposed model is an error-tolerant model that will not consider small overlaps between concepts as disagreements. This model is separated in five main phases that characterize the progression of the interactions during the naming game, including argumentation between our agents when they arise, from the presentation of an example to reaching mutual intelligibility over a part of the overall context. The first phase is the naming of an example and determining naming success. The second phase is the identification of the connected set of disagreement to which an identified disagreement belong. The third phase is the identification of disagreements trough the exchange of r-triplets and inference of overall pairing relations. The fourth phase is the resolution of each type of disagreement. The fifth optional phase is the creation of an intelligible vocabulary.

Chapter 9

An Exemplification of the Approach

9.1 General Setup

In order to illustrate the different mechanisms presented in our approach, we propose an exemplification of the process of argumentation over the meaning, described step by step from the set-up of the first encounter of the two agents to the attainment of mutual intelligibility. Before entering the details of the argumentation, we will briefly introduce the set-up of our example. This example follows an error-free model, with which both augmentation strategies are exemplified. In this exemplification, two agents A_1 and A_2 will exchange arguments in order to reach mutual intelligibility over the domain of animals.

9.1.1 Data-Set

The domain of argumentation is Zoology. in this domain, different species (examples) are regrouped in different classes that share similar properties. However, while the two agents are classifiers over the same domain, they do not have knowledge over the same species, and they do not regroup them in similar classes. Each specie e_x is represented by an array of 12 attribute/value pairs $e_x = \{a_1 = v_1, \ldots, a_{12} = v_{12}\}$ where each attribute is a Boolean variable, informing if the species displays this attribute (value = 1) or not (value = 0). The domain has a total of twelve attributes: *6_legs*, *vertebra*, *eggs*, *warm_blood*, *fly*, *carnivorous*, *terrestrial*, *social*, *4_legs*, *nocturnal*, *tail*, *brown*, *gray*, *transparent*, *small*.

Each class of species is associated to a particular set of attribute values. This set will later be the intensional definition of this class associated concept. The species of the overall context are listed in Table 9.1, and are regrouped by shared attributes in Figure 9.1.

The agents do not have access to the overall context U_O, but to two subsets of U_O —the local contexts U_1 and U_2. These two sets contain the examples:

- $U_1 = \{e_1, \ldots, e_8, e_{13}, \ldots, e_{16}, e_{21}, \ldots, e_{24}\}$
- $U_2 = \{e_5, \ldots, e_{24}\}$

Before starting the argumentation, the agents already have initial contrast sets that partition their local contexts. A_1 partitions the context U_1 with the contrast set $K_{i,1} = (S_{i,1}, U_1)$ and A_2 partitions the context U_2 with the contrast set $K_{i,2} = (S_{i,2}, U_2)$. In the sets S_1 and S_2, the agents have different concepts. Each concept has a sign, an extensional definition that is a

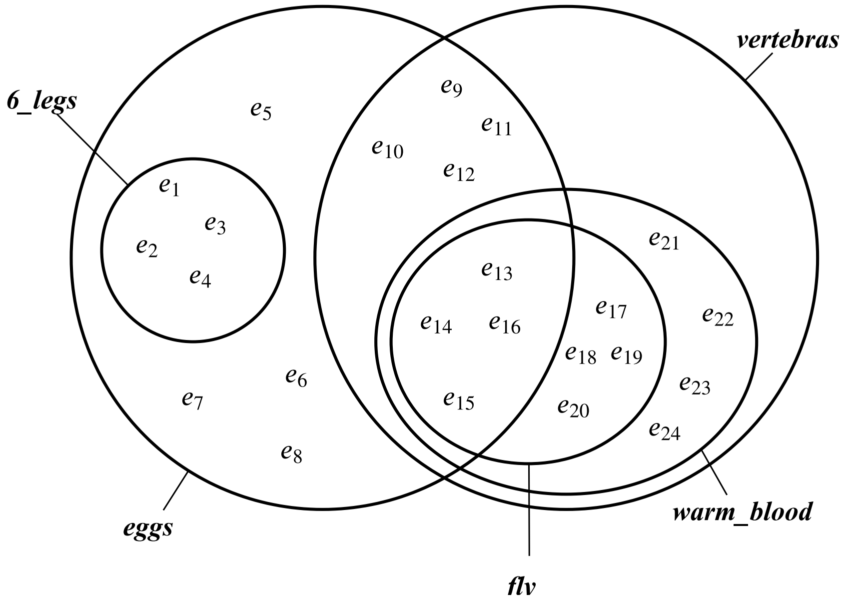


Figure 9.1: The examples of the overall context. The diagram represents which examples display each attribute. The circles are labeled with attributes, and contains the examples that display their labeled attribute. The species corresponding to each example is detailed in Table 9.1.

e_1 = stick	e_5 = lobster	e_9 = gecko	e_{13} = tit	e_{17} = brown-bat	e_{21} = tiger
e_2 = ant	e_6 = crab	e_{10} = snake	e_{14} = eagle	e_{18} = pipistrelle	e_{22} = dog
e_3 = louse	e_7 = shrimp	e_{11} = turtle	e_{15} = owl	e_{19} = hoary bat	e_{23} = whale
e_4 = flea	e_8 = crayfish	e_{12} = iguana	e_{16} = crane	e_{20} = myotis	e_{24} = horse

Table 9.1: The 24 examples representing the overall context of our exemplification. They are divided into six classes that the agents are unaware of; from left to right: crustaceans, insects, lizards, birds, bats and non-flying mammals.

subset of examples from U_O, and an intensional definition. The intensional definition contains one or more generalization(s) over the examples of the extensional definition, and are represented as one or multiple set(s) of attribute/value pairs $\{a_i = v_i, \ldots, a_n = v_n\}$ where $n \leq 5$. With this notation, we consider that a generalization $g = \{a_1 = v_1, \ldots, a_n = v_n\}$ subsumes an examples $e = \{a_1 = v'_1, \ldots, a_{12} = v'_{12}\}$ if, for all $x \in \{i, \ldots, n\}$, $v_x = v'_x$.

The agent A_1 has four concepts in its initial contrast set:

- $C_1 = \langle$*arthropod*, $\{$*6_legs=1*, *eggs=1*, *vertebra=0*$\}$, $\{e_1, \ldots, e_4\}\rangle$,
- $C_2 = \langle$*crustacean*, $\{$*6_legs=0*, *eggs=1*, *vertebra=0*$\}$, $\{e_5, \ldots, e_8\}\rangle$,
- $C_3 = \langle$*bird*, $\{$*vertebra=1*, *fly=1*$\} \vee \{$*eggs=1*, *vertebra=1*$\}$, $\{e_{13}, \ldots, e_{16}\}\rangle$,
- $C_4 = \langle$*mammal*, $\{$*vertebra=1*, *fly=0*, *warm_blood=1*$\}$, $\{e_{21}, \ldots, e_{24}\}\rangle$,

while the agent A_2 has three concepts in its initial contrast set:

- $C_5 = \langle$*arthropod*, $\{$*6_legs=0*, *eggs=1*, *vertebra=0*$\}$, $\{e_5, \ldots, e_8\}\rangle$,
- $C_6 = \langle$*reptile*, $\{$*eggs=1*, *vertebra=1*, *fly=0*$\}$, $\{e_9, \ldots, e_{12}\}\rangle$,
- $C_7 = \langle$*warm_blood*, $\{$*vertebra=1*, *warm_blood=1*$\}$, $\{e_{13}, \ldots, e_{24}\}\rangle$.

9.2 Systematic Strategy

In this section, we will present how the agents would reach mutual intelligibility on the present scenario by following a systematic strategy. The different phases, steps and states followed by the agents are the same as the ones presented in Chapter 7.

9.2.1 Starting the argumentation

Phase 1

The first action that the agents take upon starting the argumentation is the creation of a new contrast set. The agent A_1 builds a new contrast set K_1 which is a copy of $K_{i,1}$, while the agents A_2 builds a new contrast set K_2 which is a copy of $K_{i,2}$. Then, as they are in a systematic strategy, the agents look for all the pairing relations of the argumentation. In order to find the pairing relations between the concepts that are not from the same contrast set, the agents are sending to each others the sign and the intensional definition of each of their concepts. A_1 sends the pairs of semiotic elements $s(C_1), I(C_1), \ldots, s(C_4), I(C_4)$ to A_2 while A_2 sends the pairs of semiotic elements $s(C_5), I(C_5), \ldots, s(C_7), I(C_7)$ to A_1.

The agents will create their own representations of each others contrast set by building hypotheses. The agent A_1 builds a hypothesis H_1 that will contain a set of concepts $C_5^1, \ldots, C_7^1$, that will share the same signs and intensional definitions as $C_5^2, \ldots, C_7^2$ but with extensional definitions replaced by adjunct sets based on U_1, A_1's local context. In a similar fashion, A_2 builds a hypothesis H_2 that will contain a set of concepts $C_1^2, \ldots, C_4^2$.

Phase 2

Using their contrast sets and hypotheses, the agents are computing the local pairing relations. Since this is the first time they are computing pairing relations, they also consider the pairing relations between their own concepts. The two agents have access to the same sign and same intensional definitions; however, the fact that they have different contexts makes their local pairing relations sometimes different. These differences are illustrated in Figure 9.2. In order to settle on a pairing relation for each pair of concepts, the agents are going to replace their local relations by overall pairing relations.

The overall pairing relations are the same for each agent, and unify the perspectives that the agents have on the current situation of the argumentation. The overall pairing relations are represented in Figure 9.3. We can see that several pairing relations are causing disagreements. The agents will list these disagreements: since they see the same overall pairing relations, they will list the same disagreements. They identify the following:

- one self-disagreement between two concepts of A_1 (C_3 and C_4),
- three semantic disagreements (one overlap between C_4 and C_7, plus two hypo/hypernymies between C_3 and C_6, and C_4 and C_7), and
- one lexical disagreement between C_2 and C_5.

As an experimenter, we can already notice that there is an unlisted disagreement: the concept C_2 does not find an equivalent in A_2's concepts, and therefore causes an untranslatable disagreement. However, the untranslatable disagreements will be addressed later, once the agents are sure that their partitions do not need to be changed anymore. Indeed, the creation – and deletion – of

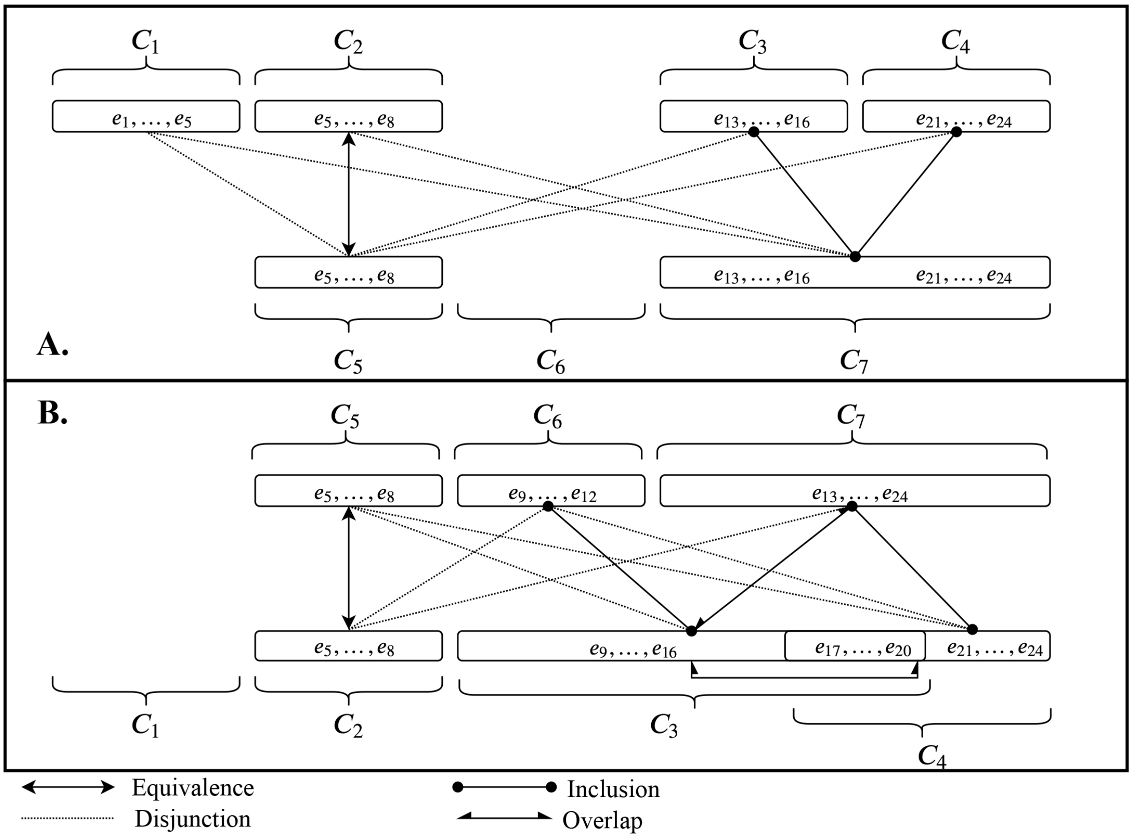


Figure 9.2: Part A represents the local pairing relations of agent A_1 while Part B represents the local pairing relations of agent A_2. The different types of pairing relations are represented by different arrows, presented below the figure.

some concepts caused by changes in the agents' partitions can incidentally resolve some of the untranslatability disagreements.

The agents end the second phase of the strategy by ensuring that there is no loss of transitivity. As we can observe in Figure 9.3, there is no concept in either contrast set that is in a relation of equivalence with two nonequivalent concepts. Therefore, the relation of equivalence is still transitive and our agents can proceed to the next phase of the argumentation.

9.2.2 Resolving the Self-Disagreement

Phase 3

The agents start to look for untranslatable disagreements. Since the argumentation just started, the agents have a lot of concepts without equivalent in the other agent's contrast set. For this reason, the concepts C_1, C_3, C_4, C_6 and C_7 are all involved in untranslatable disagreements that is added to the list of disagreements D of the agents.

The first disagreement that the agents will attempt to resolve is, as explained in Section 5.5, the self-disagreement between C_3 and C_4. Before starting to engage the disagreement that occurs between concepts of two different contrast sets, the agents need to make sure that the starting point of the argumentation is indeed a situation with two contrast sets, and at the moment K_1 is not a contrast set as two of its concepts are overlapping.

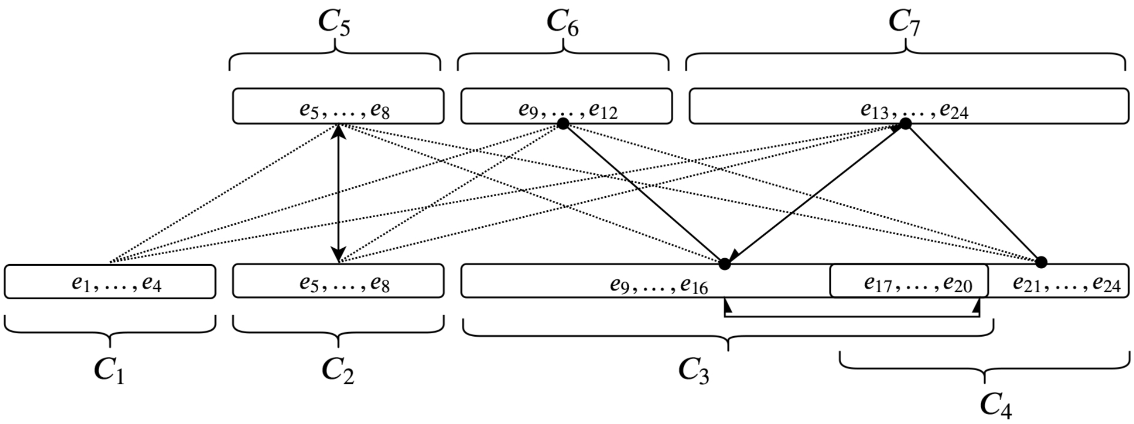


Figure 9.3: Overall pairing relations between the agents before disagreement resolutions. The legend of this figure is the same as for 9.2.

AU distance	{*vertebra=1, fly=1*}	{*eggs=1, vertebra=1*}	average $I(C_3)$	$I(C_4)$
$\{e_{17}, \ldots, e_{20}\}$	0.684	0.737	0.711	0.650

Table 9.2: Anti-unification distances between each examples from the set $\{e_{17}, \ldots, e_{20}\}$ and the generalizations (isolated and average) of the intensional definitions $I(C_3)$ and $I(C_4)$.

The agent A_1 will need to clarify the border between the concepts C_3 and C_4. A_1 does not have examples to redistribute as $Adj(C_3, U_1) \cap Adj(C_4, U_1)$ is empty; however, the agent A_2 splits the set of examples $Adj(C_3, U_2) \cap Adj(C_4, U_2)$ into two groups. We can see in Table 9.2 the similarity between each of these examples and the two intensional definitions $I(C_3)$ and $I(C_4)$. These similarity will decide to which concept should belong each example in the overlap. After this, A_1 will create, helped by A_2, two new intensional definitions I'_3 and I'_4 such that:

- I'_3 = {*eggs=1, vertebra=1, fly=1*} subsumes both $Adj(C_3, U_O) - Adj(C_4, U_O)$ and {}, and
- I'_4 = {*eggs=0, vertebra=1*} subsumes both $Adj(C_4, U_O) - Adj(C_3, U_O)$ and {}.

Once this is done, A_1 builds two new concepts:

- $C_8 = \{s(C_3), I'_3, Adj(I'_3, U_1)\}$ and
- $C_9 = \{s(C_4), I'_4, Adj(I'_4, U_1)\}$.

The concepts C_8 and C_9 come in replacement of C_3 and C_4, C_3 and C_4 are removed from K_1. In order to keep its hypothesis up to date, A_2 removes C_3^2 and C_4^2 from H_2 while adding C_8^2 and C_9^2. Both agents can now remove the disagreements that involved C_3 and C_4. These disagreements are the self-disagreement involving both C_3 and C_4, but also the three semantic disagreements and the untranslatable disagreements cause by C_3 and C_4.

Phase 2

Since the disagreement was a self-disagreement located in A_1, the agent A_1 is adding the concepts C_8 and C_9 to K_1 while the agent A_2 adds its own versions C_8^2 and C_9^2 to K_2. The agents will compute the pairing relations involving the new created concepts, C_3 and C_4. Figure 9.4 represents the point of view of both agents after finding the overall pairing relations. The resolution of

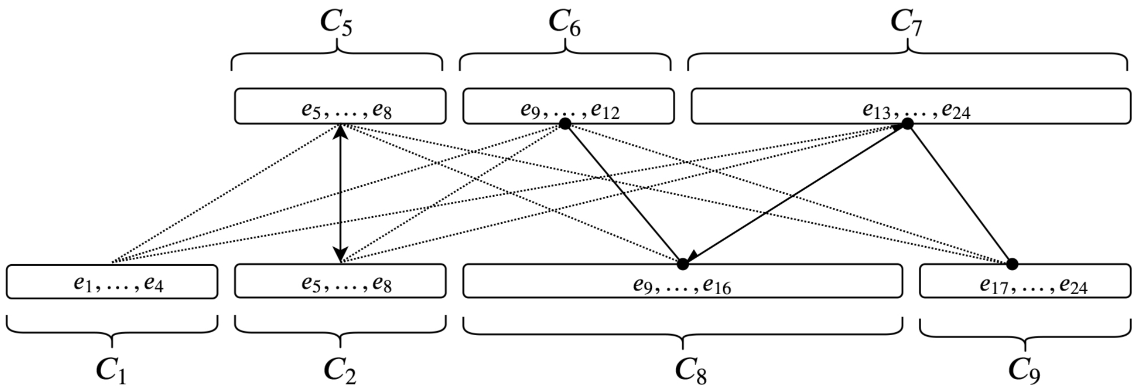


Figure 9.4: Overall pairing relations after the resolution of the self-disagreements.

the self-disagreement caused new disagreements: while removing C_3 and C_4 resolved the overlap disagreement that C_3 had with C_7 and the hypo/hypernymy disagreements that C_4 had with C_7 and C_3 had with C_6, the concept C_8 is now overlapping with C_7 and is a hypernym of C_6, while C_9 is now a hyponym of C_7.

Regarding the transitivity loss, we can observe in Figure 9.4 that once again the agents preserved the transitivity of the equivalence pairing relation. The agents can safely move to the resolution of another disagreement.

9.2.3 Resolving the Overlap Disagreement

Phase 3

The agents add the two new untranslatable disagreements caused by C_8 and C_9 and move to the selection of a new disagreement. In the absence of self-disagreement, the agents look for a semantic disagreement and find the overlap caused by C_7 and C_8. In order to resolve this disagreement, the agents create a new concept C_{10} that subsumes the intersection of $Adj(C_7, U_O)$ and $Adj(C_8, U_O)$. The creation of C_{10} is done through argumentation such that $I(C_{10}) = \{$*eggs=1*, *vertebra=1*, *fly=1*$\}$. Since the disagreement is an overlap, the new concept is added to both contrast sets —while no existing concept is removed.

Phase 2

The agent A_1 adds C^1_{10} to K_1 and H_1, and the agent A_2 adds C^2_{10} to K_2 and H_2. Upon computation of the overall pairing relations, the agents observe the situation reported in Figure 9.5. We can see that the addition of C_{10} to K_1 and K_2 causes two new disagreements: the concept C^1_{10} is a hyponym of C_7, while the concept C^2_{10} is a hyponym of C_8. The two new hypo/hypernymies are added to the list of disagreements D. After validating the transitivity of the equivalence pairing relation, the agents move to their next disagreement.

9.2.4 Resolving the Hypo/Hypernymy Disagreements

Phase 3

Since a version of C_{10}, either C^1_{10} or C^2_{10}, is added to each contrast set, no new untranslatable disagreement is added to D as the two versions of C_{10} are in an equivalent pairing relation from

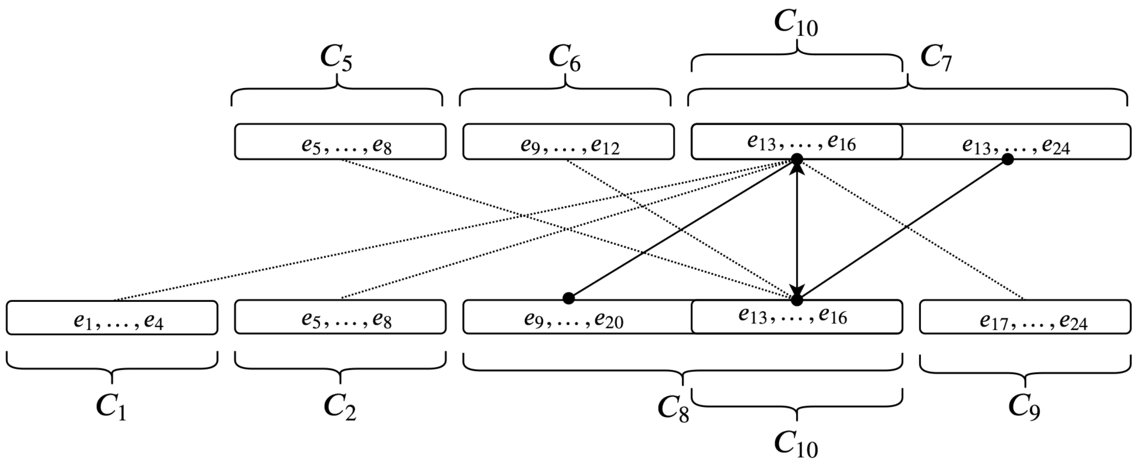


Figure 9.5: Overall Pairing Relations of the two new concepts C_{10}, created through argumentation to resolve an overlap disagreement.

their shared intensional definition. The agents move directly to the next semantic disagreement. With no more overlap disagreements, the agents are looking for a hypo/hypernymy to resolve. There are four hypo/hypernymy disagreements in the current argumentation, and the agents select one randomly, say they select $d = (s(C_{10}), s(C_8), C_{10}^2 \odot_O C_8)$.

In the disagreement d, the concept C_{10} is the hyponym and the concept C_8 the hypernym. The agents will therefore create a new concept C_{11} that will be the co-hyponym of C_{10} with regard to C_8. The agents create the new concept C_{11} through argumentation, such that its intensional definition $I(C_{11})$ is equal $\{$*eggs=1*, *vertebra=1*, *fly=0*$\}$. After the creation of the new concept, the hypernym C_8 is removed from the argumentation. The agent A_1 removes C_8 from its contrast set K_1, while the agent A_2 removes C_8 from its hypothesis H_2. The removal of C_8 resolves the related untranslatability disagreements, and also the hypo/hypernymy disagreement between C_8 and C_6 in addition to d.

Phase 2

Since A_1 has removed the hypernym of d from its contrast set, A_1 replaces it by adding the two co-hyponyms C_{10} and C_{11} to K_1. The other agent A_2 adds C_{10} and C_{11} to its hypothesis H_2. Since the disagreement resolution is compartmentalized, neither A_1 nor A_2 reacts to the fact that C_{10} was already in their contrast set and hypothesis. The agent A_1 has therefore the concept C_{10} twice in its contrast set, while the agent A_2 has C_{10} twice in its hypothesis.

Once the new concepts have been added, the two agents proceed to find their overall pairing relations with the old concepts. The new overall pairing relations are represented in Figure 9.6. The creation of the new concepts caused two new disagreements. The new concept C_{10} added to the contrast set K_1 is in a hypo/hypernymy disagreement with C_7. The second disagreements comes from the concept C_{11} that is in an equivalence pairing relation with C_6 while not sharing the same sign, therefore causing a synonymy which is immediately added to D.

While the transitivity of the equivalence relation is still respected in our argumentation, the agents have multiple equivalent concepts in their containers. Since the two versions of C_{10} in K_1 share their signs, their intensional and their extensional definitions, they are also equivalent and the agent A_1 removes the oldest version of C_{10} from K_1. The agent A_2 aligns on that decision and remove the oldest concept C_{10} from H_2. The hypo/hypernymy disagreement involving the oldest C_{10} is removed from D.

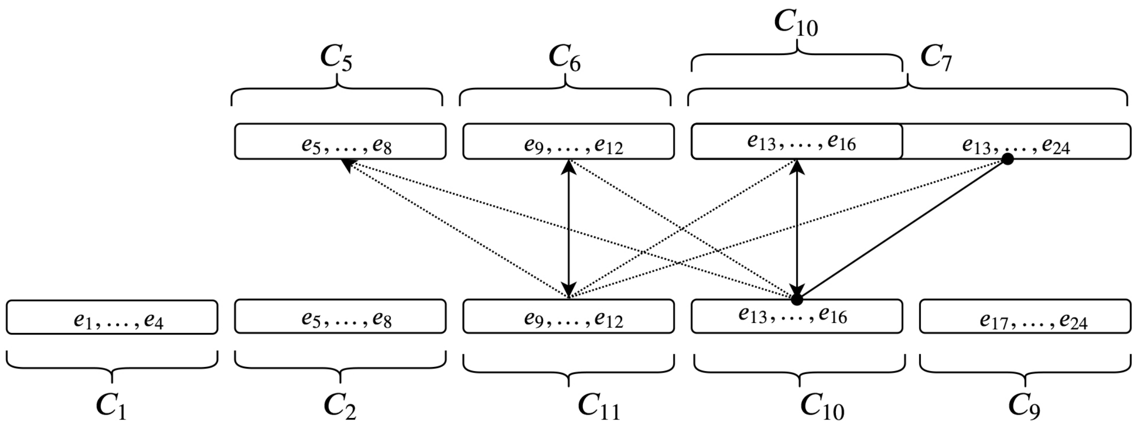


Figure 9.6: Overall Pairing Relations of the two new concepts C_{10} and C_{11}, created through argumentation to resolve a hypo/hypernymy disagreement.

Phase 3

Before choosing a new disagreement to resolve, the agents check the overall pairing relations computed in Phase 2 for untranslatability disagreements, but this time both C_{10} and C_{11} have equivalents in the other contrast set. The agents are left with two hypo/hypernymy disagreements to pick from, and randomly select the disagreement $d' = (s(C_7), s(C_9), C_7 \odot_O C_9)$.

In the disagreement d', the concept C_9 is the hyponym and the concept C_7 is the hypernym. The agents will therefore create a new concept C_{12} that will be the co-hyponym of C_9 with regard to C_7. The agents create the new concept C_{12} through argumentation, such that its intensional definition $I(C_{12})$ is equal $\{$*eggs=1*, *vertebra=1*, *warm_blood=1*$\}$. After the creation of the new concept, the hypernym C_7 is removed from the argumentation. The agent A_2 removes C_7 from its contrast set K_2, while the agent A_1 removes C_7 from its hypothesis H_1. The removal of C_7 resolves the related untranslatability disagreements, and also the hypo/hypernymy disagreement between C_7 and C_{10} in addition to d'.

Phase 2

The agent A_2 has removed the hypernym of d' from its contrast set and replaces it by adding the two co-hyponyms C_9 and C_{12} in K_2. The other agent A_1 adds C_9 and C_{12} to its hypothesis H_1. Once the new concepts have been added, the two agents proceed to find their overall pairing relations with the old concepts. The new overall pairing relations are represented in Figure 9.7. The creation of the new concepts caused a new disagreement: C_{12} is in an equivalence pairing relation with C_{10}^1 while not sharing the same sign, therefore causing another synonymy which is immediately added to D.

The transitivity of the equivalence relation continues to be respected in our argumentation, but the agents have multiple equivalent concepts in their containers. Since the concepts C_{12} and C_{10}^2 are equivalent and in a same container, the agent A_2 removes C_{10} from K_2 as it is the oldest concept. The agent A_1 aligns on that decision and removes C_{12} from H_1.

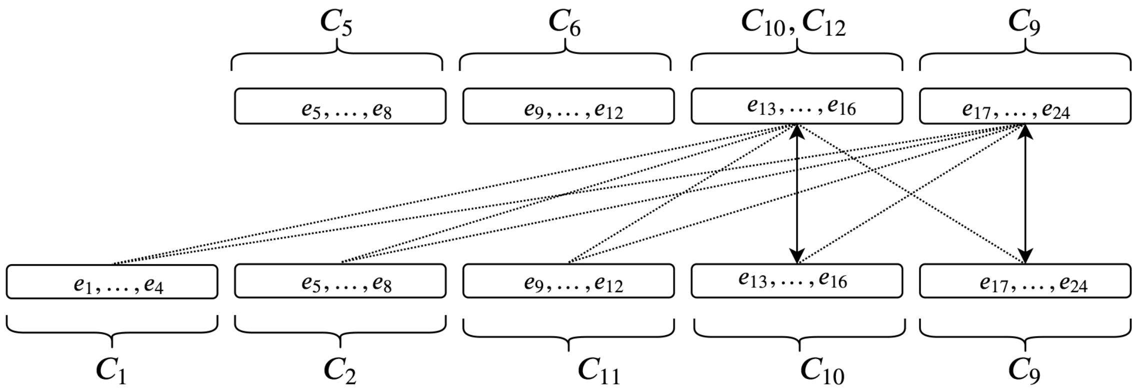


Figure 9.7: Overall Pairing Relations of the two new concepts C_9 and C_{12}, created through argumentation to resolve a hypo/hypernymy disagreement.

9.2.5 Resolving the Untranslatable Disagreement

Phase 3

This time, the agents do not have new untranslatable disagreements to resolve. The agents proceed with the argumentation by looking for a new disagreement. All the semantic disagreements being resolved, the agents are looking for untranslatable disagreements. While the agents originally listed multiple untranslatable disagreements, most of the concepts now have another concept in the contrast set of the other agent with which they have an equivalence pairing relation. The only remaining untranslatable disagreement is caused by the concept C_1 not having an equivalent in K_2. In order to resolve the untranslatable disagreement, the agent A_2 creates a concept $C_{13} = \langle s(C_1), I(C_1), Adj(C_1, U_2)\rangle$.

Phase 2

The agent A_2 adds the concept C_{13} to its contrast set K_2 while the agent A_2 adds the concept C_{13}^1 to its hypothesis. While computing the overall pairing relations of the new concept C_{13}, the agents notice that C_{13} is equivalent to C_1 (they share the same intensional definition), but has a different sign. The agents add the synonymy disagreement caused by C_1 and C_{13} to the list of disagreements D. The argumentation kept the transitivity of the equivalence relation and there are no equivalent concepts within a same container, therefore the agents continue their argumentation.

9.2.6 Resolving the Homonymy

Phase 3

The agents do not find new untranslatable disagreements, and therefore move to the resolution of lexical disagreements. There are four synonymies and one homonymy disagreement listed in D, and represented in Figure 9.8. The agents randomly select the homonymy disagreement between C_1 and C_5 as the first lexical disagreement to resolve. The agents create two new signs s_{14} and s_{15}, replace the sign of C_1 by s_{14} and the sign of C_5 by s_{15} in all containers. This operation creates two new concepts:

- $C_{14}^1 = \langle s_{14}, I(C_1), Adj(C_1, U_1)\rangle$, and

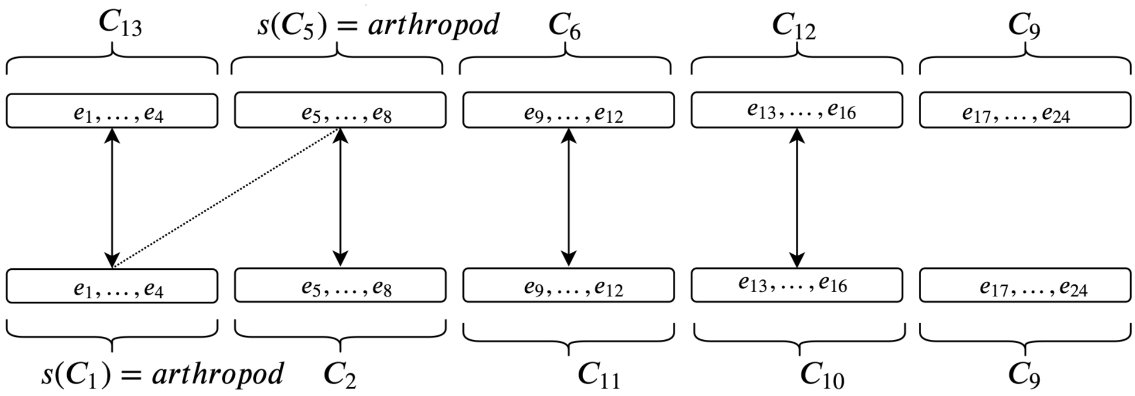


Figure 9.8: Five lexical disagreements – four synonymy disagreements and one homonymy disagreement.

- $C^2_{15} = \langle s_{15}, I(C_5), Adj(C_5, U_1)\rangle$,

and removes the two concepts C_1 and C_5 from the containers. Removing C_1 and C_5 resolves the homonymy disagreement, which is removed from D. The two synonymies that involved C_1 or C_{15} are also removed from D, but C_{14} is in a relation of equivalence with C_{13} and C_{15} is in a relation of equivalence with C_2. None of these concepts are sharing their signs, so two new synonymy disagreements are added to D.

Phase 2

Since the signs have been substituted, the agents do not need to add or remove any concepts. As there are no concepts added to Add_K and Add_H, the agents do not need to investigate any new overall pairing relations and do not need to look for new disagreements, nor do they need to check the validity of the transitivity of their equivalence pairing relation or check if an equivalence pairing relation appeared within a same contrast set. The agents can directly move forward to the next disagreement resolution.

9.2.7 Resolving the Synonymies

Phase 3-2 (x4)

The agents are left with five synonymies to resolve. The agents randomly select the synonymy disagreement between C_{14} and C_{13} as the first synonymy to resolve. The agents create a new sign s_{16} and replaces the signs of C_{14} and C_{13} by s_{14}. This operation creates a new concept in each contrast set: C^1_{16} in K_1 and C^2_{16} in K_2. Since the new concepts are the same as the older except from their signs, they keep their pairing relations and therefore the agents do not check for new disagreements. Since the two concepts that were synonyms have been replaced, the disagreement caused by C_1 and C_{13} is removed from D.

The agents repeat this operation for the disagreements:

- $ds_1 = C_2 \equiv_O C_{15}$, resulting in the creation of C_{17}
- $ds_2 = C_6 \equiv_O C_{11}$, resulting in the creation of C_{18}
- $ds_4 = C_{10} \equiv_O C_{12}$, resulting in the creation of C_{19}.

	arthropod	crustacean	bird	mammal	reptile	warm_blood
C_{16}	1	0	0	0	0	0
C_{17}	1	1	0	0	0	0
C_{18}	0	0	0	0	1	0
C_{19}	0	0	1	0	0	0.33
C_9	0	0	0	1	0	0.66

Table 9.3: Election of a new vocabulary.

Once all of these synonymy disagreements have been resolved, the agents do not have any remaining disagreements. The agents verify one last time that the transitivity of their equivalence relations holds, since there are no pairs of concepts from a same contrast set that are both in an equivalence relation with one concept of another contrast set. Furthermore, in both contrast sets, the agents observe that there are no pairs of equivalent concepts. Therefore, in the absence of synchronic disagreements, the agents conclude that they have reached mutual intelligibility.

9.2.8 Updating the Vocabulary

Phase 3

Since the set of disagreements D is now empty, the agents move directly to the fourth phase, where they will try to reuse their old vocabulary on the new contrast sets.

Phase 4

In the fourth phase, the agents create a vote for each of the signs in their old contrast set vocabularies. These votes are represented in Table 9.3. Once the votes have been exchanged and aggregated, the agents are electing the new concept that will receive each old sign. We can observe in Table 9.3 the aggregated votes and the winner concepts. After updating the signs, the agents have two contrast sets with the following concepts:

- C_{16} ($s(C_{16}) = \textit{arthropod}$)
- C_{17} ($s(C_{17}) = \textit{crustacean}$)
- C_{18} ($s(C_{18}) = \textit{reptile}$)
- C_{19} ($s(C_{19}) = \textit{bird}$)
- C_9 ($s(C_9) = \textit{mammal}$)

which grant A_1 and A_2 mutual intelligibility, reuse A_1 and A_2's old vocabulary, and are able to discriminate the same groups of examples as well as any of the initial contrast sets did.

9.3 Lazy Strategy

The initial concepts of our agents can be divided in two sets, where the concepts of one set are involved in a different set of connected disagreements than the concepts of the other set. These two sets are $\{C_1, C_2, C_5\}$ and $\{C_3, C_4, C_6, C_7\}$. If the agents are only using one of these two sets in there communication, they should not need to seek mutual intelligibility over the other set. We will see that the lazy strategy will allow them to only selectively align their concepts over one set of concepts in the same scenario.

9.3.1 Before the Naming game

Before starting the naming game, each agent create a new contrast set. The agent A_1 builds a new contrast set $K_{c,1}$ which is a copy of $K_{i,1}$, while the agents A_2 builds a new contrast set $K_{c,2}$ which is a copy of $K_{i,2}$. These two contrast sets are used to play the naming game. The two agents also create two *temporary* contrast sets, in order to argue over the meaning of their own concepts: A_1 creates the temporary contrast set $K_{t,1}$ while the agent A_2 creates the temporary contrast set $K_{t,2}$. In order to create local copies of the concepts of other agents during the argumentation, the agents also create two hypotheses: A_1 creates the current hypothesis $H_{t,1}$ while the agent A_2 creates the current hypothesis $H_{t,2}$. All the temporary containers have the local context of their agent, and do not contain any concept for the moment.

9.3.2 Starting the Naming Game

Phase 1.a

The argumentation starts with a single example presented to the two agents. In this exemplification, the first example presented to the agents is the example e_{22}, which describes the *dog* species. The agent A_1 looks for concepts in its current contrast set that subsumes the example e_{22}, and only finds the concept C_4. The agent A_1 therefore names e_{22} with the sign $s(C_4)$, which is *mammal*. The other agent A_2, has also one concept in its contrast set that subsumes the example e_{22}, concept C_7, and therefore the agent A_2 names the example e_{22} with the sign $s(C_7)$, which is *warm_blooded*. Upon exchanging the signs that they chose to name the example e_{22}, the two agents both notice that *mammal* and *warm_blooded* are two different signs.

The agents verify that neither C_4 nor C_7 have been investigated for disagreements before, which is the case. This means that the agents cannot conclude that their different naming of the example e_{22} falls under the acceptable degree of error τ that has been fixed. They will have to proceed to an argumentation involving C_4 and C_7 in order to sort out if these two concepts are causing one or more disagreements. In order to determine if the two concepts C_4 and C_7 are causing a disagreement, the agents need to find their pairing relation, which requires both agents to have access to their intensional definitions. Since the agents have not exchanged intensional definitions yet, the agent A_1 sends the pair of semiotic elements $s(C_4), I(C_4)$ to A_2, while A_2 sends the pair of semiotic elements $s(C_7), I(C_7)$ to A_1.

9.3.3 Searching the Connected Set of Disagreements

Phase 1.b – First time

The agents have now received the pairs of semiotic elements that have been exchanged at the end of Phase 1.a. Using the method described in Section 5.1.3, the agent A_1 creates a local copy of the concept C_7 while the agent A_2 creates the local copy of the concept C_4. These two concepts are the first to be listed for argumentation. The agents do not stop there, and now look for other disagreements that would be caused by C_4 and C_7. Since the two agents do not have copies of all of each other concepts, they search for these new disagreements separately in their local contexts. The agent A_1 determines that C_7 causes two disagreements in the local context U_1:

- $d_1 = \langle$*bird*, *warm_blood*, $C_3 \otimes_{U1} C_7\rangle$, and
- $d_2 = \langle$*mammal*, *warm_blood*, $C_4 \odot_{U1} C_7\rangle$.

The two concepts C_4 and C_7 were already listed for argumentation. Their overlap disagreement caused the example e_{22} to be named differently by A_1 and A_2. However, the concept C_3 has not been listed yet. Therefore, A_1 lists C_3 for argumentation and sends the pair of semiotic elements $s(C_3), I(C_3)$ to the agent A_2. The agent A_1 will also ask A_2 to verify if C_3 does not cause disagreements in A_2's local context.

On its turn, the agent A_2 has not yet received the semiotic elements $s(C_3)$ and $I(C_3)$. The agent A_2 only examines the concepts C_4 and C_7, but only finds the disagreement d_1. Since both concepts causing the disagreement d_1 are already listed for argumentation, the agent A_2 wishes to move to the next step of the argumentation. The agent A_1, however, requested that A_2 examines the concept C_3. Therefore, the agents stay in Phase 1.b for the moment and look for new disagreements again.

Phase 1.b – Second time

The agent A_2 has not added new concepts to the argumentation. Therefore, the agent A_1 has nothing to verify and is ready to move to the next step of the argumentation. The agent A_1, however, has added the concept C_3 to the argumentation and asked to the agent A_2 to determine if the concept C_3 causes disagreements in its context U_2. With the pair of semiotic elements $s(C_3), I(C_3)$ that A_2 has received, A_2 creates a local copy of C_3 and computes the local pairing relations of C_3 with all the concepts in its current contrast set. This time, the agent A_2 identifies two local disagreements:

- $d_1' = \langle \textit{bird},\ \textit{warm\_blood},\ C_3 \otimes_{U2} C_7 \rangle$, and
- $d_3 = \langle \textit{bird},\ \textit{reptile},\ C_3 \odot_{U2} C_6 \rangle$.

The two concepts causing the disagreement d_1' are already listed for argumentation, but the concept C_6 that is involved in d_3 has not been listed yet. Therefore, A_2 enlists C_6 for argumentation and sends the sign $s(C_6)$ and the intensional definition $I(C_6)$ to the agent A_1. This time, it is the agent A_2 that asks the agent A_1 to verify if C_6 does not cause any disagreement in A_1's local context. While the agent A_1 now wishes to move to the next step of the argumentation, the agent A_2 has required A_1 to search for new disagreements caused by C_6. The agents are therefore staying in Phase 1.b in order to look for new disagreements.

Phase 1.b – Third time

The agent A_1 receives the pair of semiotic elements $s(C_6), I(C_6)$ and uses it to create a local copy of C_6. With this local copy, the agent A_1 searches for disagreements caused by C_6 and any other concept from $K_{c,1}$. A_1 finds the disagreement $d_3' = \langle \textit{bird},\ \textit{reptile},\ C_3 \odot_{U2} C_6 \rangle$, which involves the concepts C_3 and C_6 that are both already listed for argumentation. Therefore, the agent A_1 agrees to move forward in the argumentation. The agent A_2 has not been aware of any new concept listed for argumentation, and therefore also agrees to take the argumentation to the next step. Since both agents agree to proceed forward in the argumentation, the argumentation can be taken to its next step. The agents now start the transfer of the concepts listed for argumentation, namely C_3, C_4, C_6 and C_7, from their current containers to their temporary containers. Both A_1 and A_2 end Phase 1.b by removing these four concepts from their current containers.

9.3.4 Resolving the Disagreements

After going through Phase 1.b, the agents will loop through Phases 2 and 3. In our exemplification, the resolution of the disagreements will be achieved in the same order as in the exemplification of the systematic strategy. Since Phase 2 and Phase 3 are almost identical in both strategies, we will not go as much into details as for the systematic strategy in order to keep the exemplification synoptic.

Phases 2-3

During Phase 2, the agents finish the transfer of the concepts C_3, C_4, C_6 and C_7 to the temporary containers. Upon the completion of this transfer, the agents proceed to identify the disagreements that are present in these containers. The agents identify the following disagreements:

- one self-disagreement d_4 between two concepts of A_1: $\langle$*bird*, *mammal*, $C_3 \otimes_O C_4\rangle$, and
- three semantic disagreements:
  - $d_5 = \langle$*bird*, *warm_ blood*, $C_3 \otimes_O C_7\rangle$
  - $d_6 = \langle$*bird*, *reptile*, $C_3 \odot_O C_6\rangle$
  - $d_7 = \langle$*mammal*, *warm_ blood*, $C_4 \odot_O C_7\rangle$

After listing these disagreements, the agents do not identify equivalent concepts in a same temporary contrast set, or concepts that are breaking the transitivity of the equivalence overall pairing relation. Therefore, the agents start resolving the identified disagreements. The agents address first the self-disagreement d_4. They redistribute the examples of the concepts C_3 and C_4 as explained in Section 5.5 and create two new concepts, C_8 and C_9. The agents then remove the concepts C_3 and C_4 that were causing the self-disagreement from their current containers. The self-disagreement d_4 is therefore resolved, along with the other disagreements d_5, d_6 and d_7 that each involved either C_3 or C_4.

Phases 1.b to 3

The agents return to Phase 1.b in order to look for new concepts from $K_{c,1}$ and $K_{c,2}$ that would be causing disagreements with the two new concepts C_8 and C_9. Since no such concept exist, the agents return directly to Phase 2.

The agents start Phase 2 by adding the concepts C_8 and C_9 to their temporary containers in order to replace the concepts C_3 and C_4, deleted during the last Phase. After this, the agents search for new disagreements caused by either C_8 or C_9 with the other concepts of the current contrast sets. After computing the missing overall pairing relations, the agents find three new semantic disagreements:

- $d_8 = \langle s(C_8), \textit{warm\_ blood}, C_8 \otimes_O C_7\rangle$
- $d_9 = \langle s(C_8), \textit{reptile}, C_8 \odot_O C_6\rangle$
- $d_{10} = \langle s(C_9), \textit{warm\_ blood}, C_9 \odot_O C_7\rangle$

All the previous disagreements have been resolved at the end of the last Phase 3, and therefore the three disagreements d_8, d_9 and d_{10} are the only listed disagreements. Since there are no concepts that are equivalent in the same contrast set, and since the equivalence pairing relation is still transitive, the agents move toward the resolution of the next disagreement.

After resolving the unique self-disagreement, the agents move forward to the resolution of semantic disagreements. The semantic disagreement that the agents select to be resolved first is the overlap disagreement d_8 between C_8 and C_7. In order to solve this overlap disagreement, the agents create a new concept C_{10}. The concept C_{10} has as adjunct set the overlapping examples between C_8 and C_7, as explained in 5.5. Since the disagreement d_8 is an overlap disagreement, the agents are not removing any concept yet. However, the disagreement d_8 is still removed from the list of active disagreements.

Phases 1.b to 3

The agents return in Phase 1.b, but the concept C_{10} does not cause any disagreement with the concepts C_1, C_2 or C_5 that stayed in the current contrast sets. Therefore, the agents return once again directly to Phase 2.

The agents start Phase 2 by adding the concept C_{10} to their temporary containers. This time, both agents add a copy of C_{10} to their contrast sets. After computing the new overall pairing relations, the agents search for disagreements caused by C_{10}. They identify two new semantic disagreements caused by concept C_{10}:

- $d_{11} = \langle s(C_{10}), C_8, C_{10} \odot_O C_8 \rangle$
- $d_{12} = \langle s(C_{10}), warm\_blood, C_{10} \odot_O C_7 \rangle$

The two disagreements d_{11} and d_{12} are added to d_9 and d_{10} in the list of active disagreements. Since there are no concepts that are equivalent in the same contrast set, and since the equivalence pairing relation is still transitive, the agents move toward the resolution of the next disagreement.

The semantic disagreement that the agents select to be resolved next is the hypo/hypernymy disagreement d_{11} between C_8 and C_{10}^2. As explained in Section 5.5, the agent A_1 creates a concept C_{11} which is the co-hyponym of C_{10}^2 with regard to the hypernym C_8. The agent A_1 also creates a sign s_{12} and a concept C_{12}, which copies the concept C_{10} of the agent A_2 such that:

$$C_{12} = \langle s_{12}, I(C_{10}), Adj(C_{10}, U_1) \rangle$$

The concept C_8 being the hypernym in the disagreement d_{11}, the agents remove it from their temporary containers where C_{11} and C_{12} will replace it. Removing the concept C_8 from the containers immediately resolves the disagreements d_9 and d_{11}, leaving only the disagreement d_{10} and d_{12} as active disagreements.

Phases 1.b to 3

The agents return in Phase 1.b, but neither the concept C_{11} nor the concept C_{12} cause any disagreement with the concepts of the current contrast sets. Therefore, the agents return directly to Phase 2. The two concepts C_{11} and C_{12} are then added to the temporary contrast set of A_1, in order to replace the removed hypernym C_8. The agents then look for disagreements caused by C_{11} or C_{12}, and find three new disagreements:

- $d_{13} = \langle s(C_6), s(C_{11}), C_6 \equiv_O C_{11} \rangle$
- $d_{14} = \langle s(C_{12}), warm\_blood, C_{12} \odot_O C_7 \rangle$
- $d_{15} = \langle s(C_{12}), s(C_{10}), C_{12} \equiv_O C_{10} \rangle$

The active disagreements are now $d_{10}, d_{12}, d_{13}, d_{14}$ and d_{15}. The agents look for pairs of concepts that are equivalent within a same contrast set, and they find that concepts C_{10} and C_{12} in $K_{t,1}$ are equivalents. The agents remove the oldest of these two concepts, the concept C_{10}, from the contrast set $K_{t,1}$. This automatically resolves the disagreements d_{12}. The equivalence pairing relation is still transitive, and the agents move toward the resolution of the next disagreement.

The semantic disagreements that remain to be addressed are the two hypo/hypernymy disagreements d_{10} and d_{14}. The semantic disagreement that the agents select to be resolved next is the disagreement d_{10}, between the concepts C_7 and C_9. The agent A_2 creates a new concept C_{13}, co-hyponym of C_9 with respect to C_7. Along with the concept C_{13}, A_2 creates a new sign s_{14} and a new concept C_{14} such that:

$$C_{14} = \langle s_{14}, I(C_9), Adj(C_9, U_2)\rangle$$

as a local copy of the concept C_9 with a substituted name. The agents then remove the hypernym C_7 from their temporary containers, which resolves the two remaining semantic disagreements d_{10} and d_{14}.

Phases 1.b to 3

Neither the concept C_{13} nor the concept C_{14} cause any disagreement with the concepts of the current contrast sets. Therefore, the agents return directly to Phase 2. The agent A_2 adds the two concepts C_{13} and C_{14} to its temporary contrast set. The two new concepts cause two new lexical disagreements:

- $d_{16} = \langle s(C_{12}), s(C_{13}), C_{12} \equiv_O C_{13}\rangle$
- $d_{17} = \langle s(C_9), s(C_{14}), C_9 \equiv_O C_{14}\rangle$

The two disagreements d_{16} and d_{17} are added to the list of active disagreements along d_{13}. The concept C_{13} is equivalent to the concept C_{10}, which is also in the contrast set $K_{t,2}$. Since C_{13} is more recent than C_{10}, the concept C_{10} is removed from the temporary contrast set of A_2. Removing the concept C_{10} immediately resolves the disagreement d_{15}. The equivalence pairing relation is still transitive, and the agents move toward the resolution of the next disagreement.

The last listed disagreements are the lexical disagreement d_{13}, d_{16} and d_{17}, which are all synonymy disagreements. The agents create three new signs s_{15}, s_{16} and s_{17}, which are then assigned as follow:

- s_{15} is substituted to the signs of both C_6 and C_{11}.
- s_{16} is substituted to the signs of both C_{12} and C_{13}.
- s_{17} is substituted to the signs of both C_9 and C_{14}.

The concepts C_{11} and C_6 become the concepts C^1_{15} and C^2_{15}, that now have an equivalent overall pairing relation *and* share the same sign. The synonymy disagreement d_{13} is resolved. Following the same reasoning, the disagreements d_{16} and d_{17} are also resolved through the creation of the concepts C_{16} and C_{17}. The concepts C_{15}, C_{17} and C_{17} are not causing any disagreement with the concepts of the current contrast sets, and therefore the agents immediately go back to Phase 2. The concept C_{15}, C_{16} and C_{17} are also not causing any disagreement with other concepts from the temporary contrast sets. Since there is also no equivalences between concepts from a same contrast set, and since the equivalence pairing relation is still transitive in the temporary containers, the agents look for a new disagreement to resolve. Since there are no more disagreement to resolve, the agents consider that they have reached mutual intelligibility.

	bird	mammal	reptile	warm_blooded
C_{15}	0	0	1	0
C_{16}	1	0	0	0.33
C_{17}	0	1	0	0.66

Table 9.4: Election of a new vocabulary after an argumentation within the lazy strategy.

9.3.5 Updating the Vocabulary

Phase 4

The agents have reached mutual intelligibility, but the concepts C now use a vocabulary different from the initial vocabulary. While the agents cannot reuse the signs $s(C_1)$, $s(C_2)$ and $s(C_5)$ that are still used in their current contrast sets, the agents can still try to reuse the signs the concepts C_3, C_4, C_6 and C_7. They have a vote that results in the score presented in Table 9.4. After updating the signs according to their vote, the agents have two contrast sets with the following concepts:

- C_{15} : ($s(C_{15})$ =*reptile*)
- C_{16} : ($s(C_{16})$ =*bird*)
- C_{17} : ($s(C_{17})$ =*mammal*)

which grant A_1 and A_2 mutual intelligibility over a part of the overall context while reusing the elements of their initial contrast sets' vocabulary. This new contrast sets are able to discriminate the same groups of examples as the concepts C_3, C_4, C_6 and C_7 did.

9.3.6 Transferring Knowledge

Phase 5

The agents have obtained two temporary contrast sets that are free of disagreements. However, they agents still have to integrate their temporary contrast sets in their current contrast sets. The agents simply transfer all the concepts from their temporary contrast sets to their current ones. Since the agents have thoughtfully ensured that the concepts from the temporary contrast sets were not causing disagreements with the ones from the current contrast sets, the transfer is safe. The agents can now continue to play the naming game, awaiting new examples, with the assurance that examples of the overall context subsumed by either C_{15}, C_{16} or C_{17} will be successfully named.

9.4 Conclusion

The systematic and the lazy strategies have a same approach to reaching mutual intelligibility —identifying relations between concepts, translating them in disagreement, and resolving the disagreements found, but they both use different path to do so. The systematic approach is frontal, exchanging all information before the start of the naming game and resolving potential causes of disagreements upfront at the cost of risking to exchange some information that would not have been used in the naming game. The lazy strategy starts the naming game and only resolve disagreements when they are causing naming differences. While this strategy can reduce by a lot the information between the agents if the naming game takes place on a limited subset of the overall context of the agents, it comes at the risk that agents might stop their naming game often to check whether or not a difference in naming is caused by a disagreement, or by a learning error within the limits of what is accepted by the agents.

Chapter 10

Experimental Evaluation

10.1 Presentation of the Experiments

10.1.1 Parameters

Before introducing the variables (dependent and independent) of our experiments, we will discuss its parameters. The parameters are not tested individually for each scenario, instead they are pre-selected. There are three parameters in our model:

1. the error threshold τ_E
2. the argument acceptability aa of the ABUI algorithm that generates the generalizations
3. the redundancy r between the two initial contexts of the agents.

Error threshold: the error threshold is the parameter τ_E already presented in Chapter 6. Referred as "the degree of error tolerated" in the figures, it gives the number of examples that is required by the agents for overall pairing partial sets to be taken into account in the definition of overall pairing relations. A different error threshold is linked to every data-set, as the value of the error threshold affects the number of classes available to set up disagreements.

Argument Acceptability: the argument acceptability is a parameter of the ABUI algorithm that determines weather or not a generalization generated by ABUI is considered satisfactory. For instance, during the construction of a counter-argument α, an agent builds a set of positive and negative examples that α should and should not cover. In this situation, the argument acceptability is the accuracy of α over the sets of positive and negative examples above which the ABUI algorithm considers that α is satisfactory enough to endorse the role of a counter-argument. By default, the argument acceptability is 0.75 in our experiments, as it is the default value used in the AMAIL argumentation framework.

Redundancy: the redundancy is the percentage of each agent's initial context that is shared among them. If the agents receive 60 labeled examples at the beginning of the experiment, 30 of which are in both initial context, the redundancy is 50%. If the same 60 examples are the initial contexts of both agents, the redundancy is 100%. The redundancy is defined by the ratio of shared examples, the label of these examples are not taken into account and two agents might have $r = 100\%$ although they do not share a single right-path association. By default, the redundancy is zero in our experiments, as it is the most complex scenario four our agents to reach mutual intelligibility over. Indeed, a redundancy of 0 indicates that no information is initially shared by the agents in the form of examples.

10.1.2 Variables

This section presents the variables that will be found in our experiments. They are presented in two distinct categories: independent variables (IV) and dependent variables (DV). We have four independent variables: *domain*, *strategy*, *setup disagreements* and *number of initial concepts*. There is a total of four dependent variables: the *Synchronic Agreement Ratio*(SAR), the *Diachronic Agreement Ratio*(DAR), the *Exchanged Examples Ratio*(EER), the *Number of Expected Concepts*(NEC) and the *Number of Final Concepts* (NFC).

Independent Variables (IV)

Domain Each domain corresponds to a different data-set. The first two domains correspond to two of the three data-sets already presented in Chapter 9: *Zoology* and *Sponges*. The remaining domain *Soybean* correspond to the eponymous data-set. The Soybean data-set has 307 instances of soybean observations spread among 19 classes of soybean diseases. There are 35 categorical attributes, some nominal and some ordered. While the experiments always use a redundancy of 0% and an argument acceptability of 0.75, each domain has its associated error threshold.

Strategy Our model is divided in two strategies: the systematic and the lazy. The systematic strategy is presented in Chapter 7 and is a strategy that focuses on preventing any disagreement that could occur on the overall context of the two agents. The lazy strategy is presented in Chapter 8 and unlike the systematic strategy, it focuses on solving disagreements on connected sets of disagreements once the agents encounter an error in their naming game.

Setup Disagreements (SDC) While the actual number of disagreements that the two agents will encounter depends on the learning of their initial contrast sets, the disagreements that we expect the agents to solve are experimentally set up. The set up of these disagreements has already been discussed in Chapter 9. This set up takes two parameters: the type of disagreements (overlap, hypo/hypernymy, synonymy or homonymy) that are set up, and for each type, the number of these disagreements.

Number of Initial Concepts (OCC) The number of initial concepts is the number of categories that are present in the data-set related to the experiment's domain. For the lowest possible error threshold, one, we have 3 concepts in the *Sponges* domain, 7 concepts in the *Zoology* domain, and 19 concepts in the *Soybean* domain. However, increasing the error threshold τ_E decreases the number of initial concepts that can be used, as the initial concepts should contain at least τ_E examples.

Dependent Variables (DV)

Count of Disagreements(C:type,context) The first measure of an experiment success is the count of each type of disagreement that exist between the two agent at the beginning and at the end of an argumentation. We therefore have an initial and a final count of disagreements, for each type of disagreement: self-disagreements, overlap disagreements, hypo/hypernymy disagreements, synonymy disagreements, homonymy disagreements, indistinguishable disagreements and untranslatable disagreements. Since there is a distinction between the local and the overall disagreements, the count can be local (as one agent sees it) or overall (as the experimenter sees it). Our model stops when there is no overall disagreement detected by the agents anymore. For this reason, we always expect the final *overall* count for each type of disagreement, to be equal to zero. Other variables will give a more nuanced measure of our model success.

Synchronic Agreement Ratio (SAR) The Synchronic Agreement Ration, or SAR, is the ratio between the of examples from the overall context that are named through a left path association with the same unique sign by both agents when presented to them, over the total number of examples in the overall context. The SAR measures how well the agents have reached mutual intelligibility.

Definition 56 (SAR)**.** *Let A_1 and A_2 be two agents, and $K1$ and $K2$ be their contrast sets. The Synchronic Agreement Ratio of the two agents is:*

$$SAR(A_1, A_2) = \frac{|\{e \in U_O | e \,{}_{K1}\!\overset{l}{\mapsto} s \wedge e \,{}_{K2}\!\overset{l}{\mapsto} s\}|}{|U_O|}$$

Diachronic Agreement Ratio (DAR) The Diachronic Agreement Ratio in an agent is the additive inverse of the ratio between (1) the number of pairs of examples from an agent's initial context that were in two different concepts in the initial contrast set that later are in the same concept in the final contrast set, and (2) the number of pairs of examples that were in two different concepts in the initial contrast set. The DAR is a measure of refinement that shows how well the monotonic evolution of the contrast sets have been respected through the argumentation.

Definition 57 (DAR)**.** *Let A be an agent that has $K = (U, Q)$ for initial contrast set and $K' = (U', Q')$ for final contrast set. The Diachronic Agreement Ratio of A is:*

$$DAR(A) = 1 - \frac{|\{e_1, e_2 \in U | e_1 \,{}_{K}\!\overset{l}{\mapsto} s \wedge e_2 \,{}_{K}\!\overset{l}{\mapsto} s \wedge e_1 \,{}_{K'}\!\overset{l}{\mapsto} s \wedge e_2 \,{}_{K'}\!\overset{l}{\mapsto} s' \wedge s \neq s'\}|}{|\{e_1, e_2 \in U | e_1 \,{}_{K'}\!\overset{l}{\mapsto} s \wedge e_2 \,{}_{K'}\!\overset{l}{\mapsto} s' \wedge s \neq s'\}|}$$

Exchanged Examples Ratios (EER) The exchanged examples ratio (EER) corresponds to the number of examples that have been sent from one agent to the other through messages, divided by the number of examples in the overall context.

Coverage Ratio (CR) The coverage ratio (CR) corresponds to the number of examples from the overall context that can be associated with a sign by an agent through left-path associations, divided by the number of examples in the overall context.

Observed Disagreements Count (ODC) Unlike the set-up disagreements (SD), the observed disagreements (OD) are disagreements *stricto sensus.* They are the disagreements that are observed between the agents after they learned their initial contrast sets. Similarly to the set-up disagreements, we measure their types and number. The setup disagreements are unlikely to be the only disagreements found between our two agents once they learn they initial contrast sets, as they built their knowledge upon partial information on the overall context. ODs can be counted before or after the argumentation, and using either the local context of one agent or the overall context.

Final Disagreements Count (FDC) The final disagreements are the disagreements found at the end of the argumentation. While their count is interesting, a number of final disagreement equals to zero is synonymous of full synchronic agreement between the agents only in scenarios where we are not accounting a degree of error. When accounting a degree of error, there can be no disagreement between the agents while the agents still associate some examples with different signs (in a proportion indexed on the error threshold that is therefore fixed).

Number of Expected Concepts (NEC) The number of expected concepts represents the total number of concepts that should be expected in a contrast set that has a concept for each polylexematic class in the combined left-path associations of both agents —which corresponds to the result of a brute force approach to the problem addressed in this thesis. Given two agents A_1 and A_2 that have respectively m and n concepts in their contrast sets, their maximum number of expected concept is $2^{m+n} - 1$ concepts, which corresponds to the number of possible combinations between the agents' adjunct sets minus the empty set.

Number of Final Concepts (NFC) The number of final concepts, or NFC, is the number of concepts in the final contrast set of one agent. For an agent A with a final contrast set $K = (U, Q)$, the NFC is equal to $|Q|$. The NFC can be measured in any of the two agents, as our protocol ensures that both agents end the argumentation with the same number of concepts in their contrast sets.

10.1.3 Tested Hypotheses

Our model is evaluated through an array of hypotheses that are tested. There is a total of six tested hypotheses, each one corresponding to different combinations of experiments and testing a specific property of our model.

1. **H1: Generality** Our model allows two agents to reach mutual intelligibility without creating diachronic disagreements regardless of the type or combinations of types of disagreements between these two agents.

2. **H2: Domain Independence**: Our model allows two agents to reach mutual intelligibility without creating diachronic disagreements regardless of the domain of their overall context.

3. **H3: Coverage Preservation** Our model allows two agents to reach mutual intelligibility without restraining the context of the agents, as the overall context of the agents should not lose examples through the argumentation.

4. **H4: Efficiency** Our model allows two agents to reach mutual intelligibility without exchanging any significant portion of their contexts.

5. **H5: Scalability** A linear increase of size of the overall context does not exponentially increase the number of generalizations that need to be exchanged in order for the agent to reach mutual intelligibility.

6. **H6: Simplicity** The number of concepts in each contrast-set after that the agents have reached mutual intelligibility is the number of expected concepts NEC that can be computed before the argumentation.

10.2 Disagreement Experiment Setup

The main goal of our model is to achieve mutual intelligibility. The notion of mutual intelligibility has been conveniently translated as an absence of disagreement, and as we mentioned in the previous section it is this absence of disagreements that constitutes the stop condition of our argumentation model. In order to evaluate our model, it is imperative to setup disagreements before the agents enter their argumentation. However, setting up disagreements is non-trivial. A disagreement is based on a pairing relation, pairing relation that is based on three pairing partial sets, partial sets that are based on as many adjunct sets, adjunct sets that are based on intensional definitions. In order to generate a certain type of disagreement, we need generate a certain set of intensional definitions for the initial concepts of our agents. These concepts —and their intensional definitions— are learned over sets of right-path associations, as explained in Section 5.1.3.

We cannot talk about disagreements at the level of the right-path association sets, as disagreements are located at the intensional level and therefore appear after the creation of left-path associations during the concept creation. However, we can purely conceptually imagine that the adjunct sets of the agents are replaced by the classes of the same agents initial sets of right-path associations, which allows us to define disagreement *precursors*. The intersection and set differences of two classes are used to compute pairing partial sets precursors, used themselves to compute a r-triplet precursor which gives a pairing relation precursor that indicates —or not, depending on the r-triplet and the classes signs— a disagreement type.

Setting up disagreement precursors is straightforward. Instead of having two different data-sets as the initial sets of right-path associations on which our agents learn their first contrast sets, the agents can use two copies of a same data-set $U \mapsto S$ that are slightly modify in order to have disagreement precursors. Once a desired disagreement precursor $p = (s_1, s_2, U(\mapsto s_1) r_U U(\mapsto s_2))$ is designated, the right-path associations of each copy can be modified in order to produce two classes $U(\mapsto s_1)$ and $U(\mapsto s_2)$) that have the right signs and intersection in order to cause p. This process is detailed at the end of the paragraph for four different types of disagreement precursors.

Translating disagreement precursors into proper disagreements is however complex. In order to move from the extensional to the intensional level, the agents need to learn generalizations over their right-path associations. This is normally done during the creation of each contrast-set, using the method described in Section 5.1.3. If the inductive learning responsible for the creation of a new concept C_i manages to reach the maximal accuracy, the examples subsumed by the new concept are the same as the examples of the class $U(\mapsto s_i)$ on which the new concept is based. In other words, the adjunct set $Adj(C_i, U)$ corresponds to the class $U(\mapsto s_i)$. Therefore the pairing partial sets and their corresponding precursors are identical, the r-triplets are identical to the r-triplet precursors, and C_i's pairing relations cause disagreements that are similar to the disagreement precursors that we set up. However, we are assuming here an accuracy that is not guaranteed during the inductive learning. For this reason, the disagreements in which C_i is involved might be different from the disagreement precursors that we set up.

Another issue with this method is that our agents are indeed having different right-path associations as a result of the disagreement precursors setup, but since we are not adding or removing examples from their two copies of the set $U \mapsto S$, both agents remain with the same context U. This issue is easily overcome by removing different right-path associations from each local sets of associations. The proportion of right-path associations left in both contexts defines the redundancy of our initial contexts that has already been discussed in Section 10.1.1. During the deletion of the associations, a particular attention should be put toward having the same number of remaining examples for each initial class of $U \mapsto S$ in both agents. Unbalancing the number of examples in two copies of a class would lead to more difficulties for the inductive learning that takes place during the concepts creation. These difficulties are likely to decrease the accuracy of the inductive learning, increasing the difference between the adjunct sets of the new concepts and their corresponding classes. As we discussed in the previous paragraph, this would increase the difference between the disagreements appearing in our augmentations and their precursors.

The general idea behind disagreement setup is therefore to first set up disagreement precursors by duplicating a data-set, rearranging the classes and the signs differently in each instance of the data-set in a fashion that forces the concepts created through right-path associations to cause disagreements. Then, removing one instance of an example from one of the two contrast set obtained results in having different contexts for each agent, finalizes the disagreement setup. The following sections explains how manipulating the associations of two copies of a same set of right-path associations can set up disagreement precursors. For the rest of this section, we consider that the disagreements are setup with two sets of right-path associations $U_1 \mapsto S_1$ and $U_2 \mapsto S_2$, such that initially $U_1 \mapsto S_1 = U_2 \mapsto S_2$.

Hypo/Hypernymy disagreements In order to setup a hypo/hypernymy disagreement with the two sets of associations $U_1 \mapsto S_1$ and $U_2 \mapsto S_2$, one class of $U_1 \mapsto S_1$ should contain all of the examples of more than one another class of $U_2 \mapsto S_2$. This is done through selecting two classes $U_1(\mapsto s_1)$ and $U_1(\mapsto s_2)$ and merging them into a class $U_1(\mapsto s_3) = U_1(\mapsto s_1) \cup U_1(\mapsto s_2)$. Then, the sets of associations $U_1(\mapsto s_1) \mapsto \{s_1\}$ and $U_1(\mapsto s_2) \mapsto \{s_2\}$ are removed from $U_1 \mapsto S_1$ and replaced by $U_1(\mapsto s_3) \mapsto \{s_3\}$. This procedure, illustrated in Figure 10.1.A, sets up two hypo/hypernymy disagreements.

Overlap Disagreements In order two setup an overlap disagreement with the two sets of associations $U_1 \mapsto S_1$ and $U_2 \mapsto S_2$, one class of $U_1 \mapsto S_1$ should share only some of its examples with only some examples of another class from $U_2 \mapsto S_2$. This is done through selecting two classes $U_1(\mapsto s_1)$ and $U_1(\mapsto s_2)$ and merging them into a class $U_1(\mapsto s_3) = U_1(\mapsto s_1) \cup U_1(\mapsto s_2)$. Then, the sets of associations $U_1(\mapsto s_1) \mapsto \{s_1\}$ and $U_1(\mapsto s_2) \mapsto \{s_2\}$ are removed from $U_1 \mapsto S_1$ and replaced by $U_1(\mapsto s_3) \mapsto \{s_3\}$. This first step is similar to the setup of a hypo/hypernymy disagreement. The second step is to select the class $U_2(\mapsto s_2)$ and to merge it with another class $U_2(\mapsto s_4)$ in order to obtain the class $U_2(\mapsto s_5) = U_2(\mapsto s_2) \cup U_2(\mapsto s_4)$. Then, the sets of associations $U_2(\mapsto s_2) \mapsto \{s_2\}$ and $U_2(\mapsto s_4) \mapsto \{s_4\}$ are removed from $U_2 \mapsto S_2$ and replaced by $U_2(\mapsto s_5)$. This procedure, illustrated in Figure 10.1.B, sets up one overlap and two hypo/hypernymy disagreements.

Synonymy Disagreements In order to setup a synonymy disagreement with the two sets of associations $U_1 \mapsto S_1$ and $U_2 \mapsto S_2$, one class of $U_1 \mapsto S_1$ should share all of its examples with another class from $U_2 \mapsto S_2$, but not its sign. This is done through selecting a class $U_1(\mapsto s_1)$ and using it to create a set of right-path associations $U_1(\mapsto s_1) \mapsto \{s_2\}$. Then, the set of associations $U_1(\mapsto s_1) \mapsto \{s_1\}$ is removed from $U_1 \mapsto S_1$ and replaced by $U_1(\mapsto s_1) \mapsto \{s_2\}$. This procedure, illustrated in Figure 10.1.C, sets up one synonymy disagreement.

Homonymy Disagreements In order to setup a homonymy disagreement with the two sets of associations $U_1 \mapsto S_1$ and $U_2 \mapsto S_2$, one class of $U_1 \mapsto S_1$ should share no example with another class from $U_2 \mapsto S_2$, while sharing its sign. This is done through selecting two classes $U_1(\mapsto s_1)$ and $U_2(\mapsto s_2)$, using them to create two sets of right-path associations:

- $U_1(\mapsto s_1) \mapsto \{s_3\}$, and
- $U_2(\mapsto s_2) \mapsto \{s_3\}$.

Then, the set of associations $U_1(\mapsto s_1) \mapsto \{s_1\}$ is removed from $U_1 \mapsto S_1$ and the set $U_2(\mapsto s_2)$ is removed from $U_2 \mapsto S_2$. The set $U_1(\mapsto s_1) \mapsto \{s_3\}$ is then added to $U_1 \mapsto S_1$, while the set $U_2(\mapsto s_2) \mapsto \{s_3\}$ is added to $U_2(\mapsto s_2)$. This procedure, illustrated in Figure 10.1.D, sets up one homonymy and two synonymy disagreements.

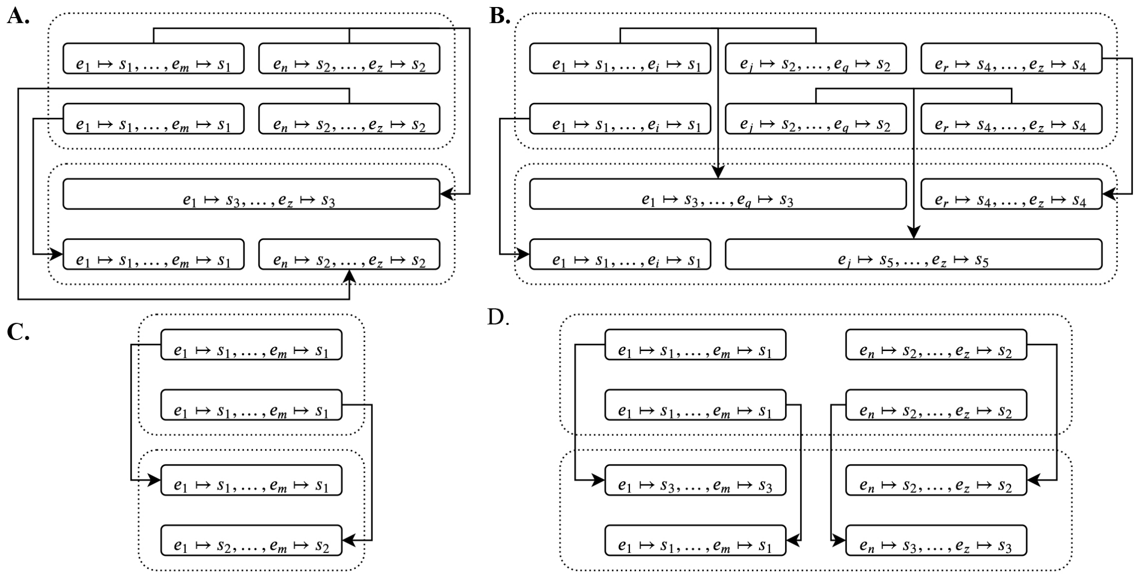


Figure 10.1: Setup of four types of disagreements. The continuous lines represent the classes from two equivalent sets of associations, while the doted lines regroup the classes before and after the setup of the disagreements. Group A shows the setup of a hypo/hypernymy disagreement, group B shows the setup of an overlap disagreement, C shows the setup of a synonymy disagreement and group D the setup of a homonymy disagreement.

10.3 Generality

The hypothesis of generality postulates that both approaches of our model can resolve each possible combination of disagreements between two agents' contrast sets, increasing their SAR while keeping their perspective DAR at one. In order to test this hypothesis, we used the Soybean data-set to setup variable disagreements between two agents. We counted the number of disagreements before and after argumentation, along with the SAR and the DAR. In order to evaluate this hypothesis on a great number of disagreements and combinations of disagreements, we have run this experiment 200 times with different arrangements.

Four types of disagreements are being setup in each run: overlap, hypo/hypernymies, synonymies and homonymies. These four types are randomly ordered for each experiment run, and a random number of each disagreement type is selected and then setup, one type after the other. For instance, if the overlap type has been ordered first, we select a random number between 0 and the number of classes of Soybean with more than τ_E examples divided by three (the number of classes required to setup an overlap disagreement) and we merge as many time three random classes in order to obtain as many overlap disagreements. For each of the 200 runs of the experiment, we setup a new random arrangement of disagreements.

The independent variables of this experiment are the types of disagreements that have been setup and their number (SDC). The experiment is always using the Soybean data-set, an error threshold of $\tau_E = 5$ and an argument acceptability of 0.75. The main dependent variables are the SAR and the DAR, in order to measure to which extend our model reached a mutual intelligibility in a monotonic way. These variables are completed by the different example counts (ODC, FDC) and the coverage ratio (CR).

The setup disagreements are unlikely to be the only disagreements found between our two agents once they learn they initial contrast sets, as they built their knowledge upon partial information on the overall context. For this reason, we also count the number of observed disagreements. The

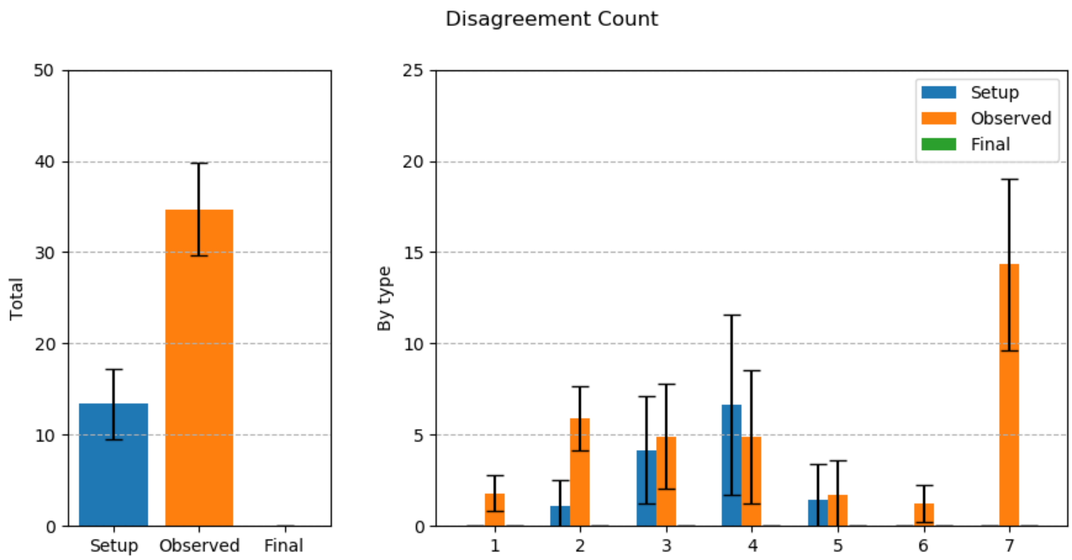


Figure 10.2: Disagreement count in a systematic argumentation with random disagreements. The left plot presents the total count of disagreements, for all their types. On the right plot, the y-axis shows the disagreement count for the types of disagreement displayed in the x-axis: 1.Self-Disagreement, 2.Overlap, 3. Hypo/hypernym, 4. Synonym, 5.Homonym, 6.Indistinguishable disagreement, 7.Untranslatable disagreement.

number of observed disagreements can be counted before or after the argumentation, and using either the local context of one agent or the overall context. The number of observed disagreements is an additional dependent variable.

10.3.1 Systematic Strategy

The results of this experimental evaluation are shown in Figures 10.2, 10.3 and 10.4 below. Figure 10.2 displays the average count of disagreements over our collection of experiments. The left figure represents the average counts of overall disagreements, while the right picture separates the average counts of disagreements according to their types. The counts are done at three different moments of the argumentation: the set up disagreements are counted *before* the argumentation, as they are the arguments that we set up in the agents learning data-sets prior to the construction of their initial contrast sets. The observed disagreements are counted *at the beginning* of the argumentation, once the initial contrast sets have been learned. The final disagreements are counted *after* the argumentation, and should therefore always be equal to 0.

In the left part of Figure 10.2, we can observe that the number of observed disagreements is more than twice the number of disagreements that have been set up. While disagreements spontaneously arise between two agents once they learned their first contrast sets over two different contexts, setting up disagreements increases the heterogeneity between the two agents' representations and causes a higher number of disagreements. The set up disagreements can be seen as “stems” for the observed disagreements. The final number of disagreements, which is counted after each argumentation, is always zero. This is not surprising, as an absence of disagreement is not necessarily equivalent to a full mutual agreement if the model admits a positive error threshold. The absence of overall disagreements is in fact the condition that tests the end of the argumentation between the agents in the systematic strategy, and therefore an equal-to-zero number of disagreements after the argumentation was expected.

The right part of Figure 10.2 breaks down the count of disagreements for each different types of

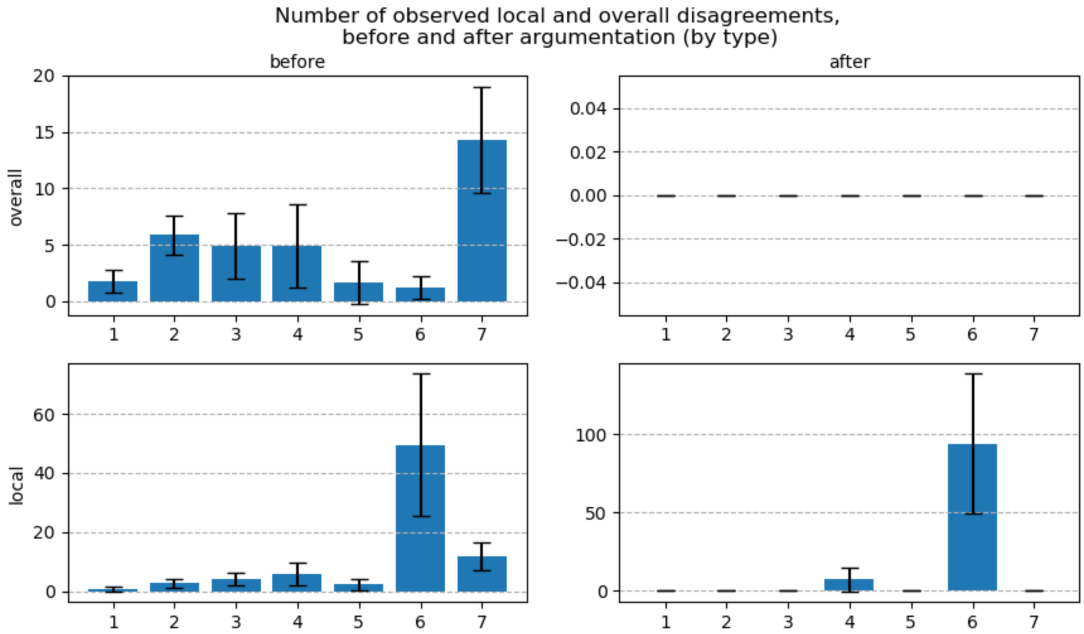


Figure 10.3: Disagreement count in a systematic argumentation with random disagreements. The y-axis displays the count of each of the types of disagreement displayed in the x-axis: 1.Self-Disagreements, 2.Overlaps, 3. Hypo/hypernymies, 4. Synonymies, 5.Homonymies, 6.Indistinguishable disagreements, 7.Untranslatable disagreements

disagreements. As mentioned before, the only set up disagreement types that have a positive count are following: overlap, hypo/hypernym, synonym and the homonym. The distribution of the set up disagreements is directed by the cost of setting up each type: for instance, setting up an overlap requires three classes from the initial data-set. On the other hand, setting up a synonymy only requires one class. Moreover, certain disagreements are causing other type of disagreements: for instance, setting up an overlap also sets up two hypo/hypernym disagreements.

The distribution of the observed disagreements is very different from the distribution of the setup disagreements. For instance, the most observed type of disagreement from the four types that are setup is the overlap disagreement, which was also the least set up type of disagreement. Indeed, since setting up an overlap requires to use 3 classes versus two at most for the other three types, they are expected to be less. The high incidence of overlap disagreements in the observed disagreements is an indicator that the overlap disagreement is the most likely to appear spontaneously when two agents learn their concepts over different contexts. Moreover, we can see that the most observed type of disagreements is the untranslatable disagreement. Since our error threshold is low ($\tau_E = 5$), and since the Soybean data-set has many small classes having about this number of examples, this is due to the agents over-fitting their intensional definitions because of the scarcity of examples in their initial data-sets.

Figure 10.3 shows the count of each type of disagreements before and after the argumentation. The top figures represent the overall disagreements, while the bottom figures represent the local disagreements of one of the agent. Since the argumentation model is symmetrical and the disagreements are set up randomly, the other agent's counts follow a similar profile. On horizontal axis, the leftmost figures display the ODC, the disagreement count at the beginning of the argumentation, while the rightmost figures display the FDC, the disagreement count at the end of the argumentation. We can observe that while their are never overall disagreements after an argumentation, there are still local disagreements. While or model focuses on resolving every overall disagreements, it does not aim to resolve the local disagreements. Indeed, local disagreements do

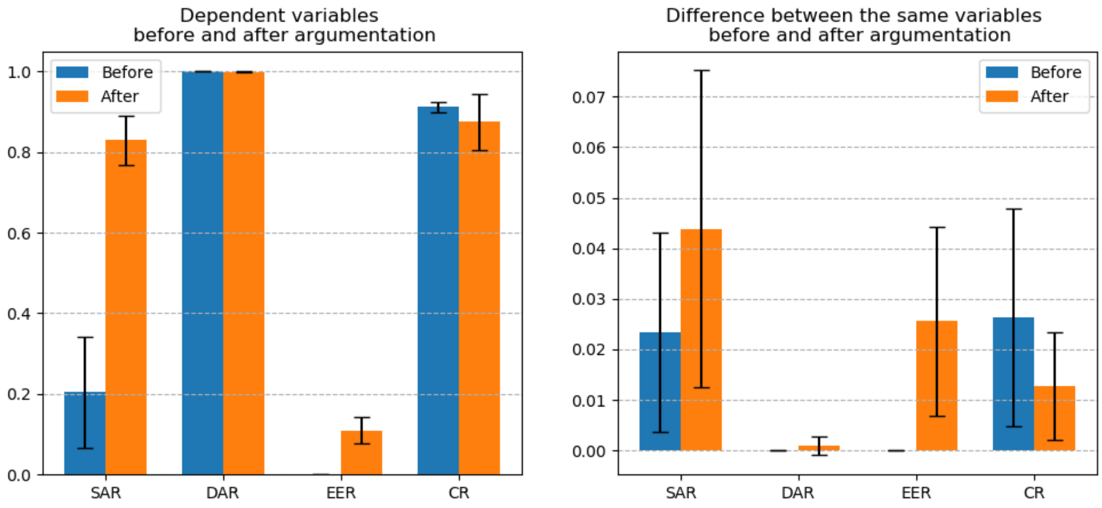


Figure 10.4: Evolution of SAR, DAR, EER and CR before and after a systematic argumentation where the disagreements are randomly selected (left). A local version (taking only the examples of local contexts) of these five variables is implemented and the the euclidean distance between the two agent's variable is measured (right).

not necessarily stop the mutual intelligibility between the agents. Moreover, once there are no more overall disagreement between the agents anymore, eliminating the remaining local disagreements would require to exchange examples. While this would give more information to the agents, allowing them to observe directly that they have reached mutual intelligibility with their contexts, it would also go against our goal to keep the transfer of examples between the agents at the lowest possible amount.

Figure 10.4 presents all the ratio that have been measured, before and after the argumentation. The most important are the SAR and the DAR, as the EER and the CR will be explored with more details in future hypothesis tests. The leftmost figure presents the overall measures. The DAR, that is in essence a local measure which compares two contrast sets of a same agent, is here presented as the average of two DARs of the agents, averaged again over the 200 experiments. The rightmost figure presents the average difference between the local measures of a same experiment. Since the SAR, EER and CR are global measures, we propose a formulation for their local counterparts. The local SAR of an agent A is computed by using the local context of A instead of the overall context as the denominator of the ratio. The local EER of A is computed by using the number of examples sent by A as the numerator of the ratio, instead of using the number of examples exchanged by the two agents which would also include the number of examples received by A. Finally, the local CR of A is computed by dividing the number of examples from A's context that are covered by one of its current contrast set's concept, by the number of examples in A's context.

We can see on the leftmost figure that the DAR of both agents is one both before and after the argumentation, meaning that the agents refined their respective contrast sets in a monotonic way, without compromising their original classifications. Moreover, the SAR significantly increases after the argumentation. The agents are therefore able to reach mutual intelligibility while refining their contrast sets in a monotonic way with our model. The EER stays around 0.1 with a low variance, meaning that the agents do not need to exchange more than 10% of their examples to reach mutual intelligibility for any type of disagreement combination encountered. The CR decreases after the argumentation, meaning that less examples are covered after the argumentation. This is coherent with the fact that our model has been design to chose refinement over coverage. However, the difference between the CR before and after the argumentation is small, with a CR after argumentation well above 0.5, which indicates that the refinement does not cause an over-

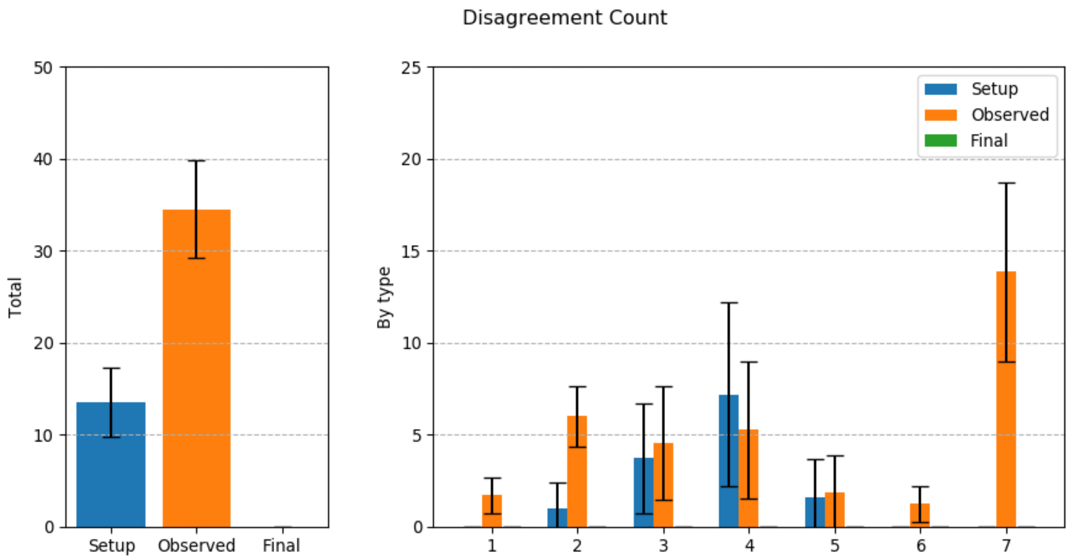


Figure 10.5: Disagreement count in a lazy argumentation with random disagreements. The left plot presents the total count of disagreements, for all their types. On the right plot, the y-axis shows the disagreement count for the types of disagreement displayed in the x-axis: 1.Self-Disagreement, 2.Overlap, 3. Hypo/hypernym, 4. Synonym, 5.Homonym, 6.Indistinguishable disagreement, 7.Untranslatable disagreement.

fitting.

The scale on the rightmost figure ranges from 0 to 0.08. This means that the difference between the local measures on the two agents ranges on a tenth of the global measures. From this, we can conclude that there is no asymmetry between the knowledge of our agents before or after the argumentation.

10.3.2 Lazy Strategy

The results of this experimental evaluation on the lazy strategy are shown in Figures 10.5, 10.6 and 10.7. Figure 10.5 displays the average count of disagreements over our collection of experiments. The left-hand plot represents the average counts of overall disagreements, while the right-hand plot separates the average counts of disagreements according to their types. The counts are done at three different moments of the argumentation: the set up disagreements are counted *before* the argumentation, as they are the arguments that we set up in the agents learning data-sets prior to the construction of their initial contrast sets. The observed disagreements are counted *at the beginning* of the argumentation, once the initial contrast sets have been learned. The final disagreements are counted *after* the argumentation, and should be equal to 0 if the process is successful.

We observe that the counted disagreements *before* and *at the beginning* of the argumentation presented in Figure 10.5 are similar to the corresponding counts presented in Figure 10.2. A similar count in setup and observed disagreements and a similar distribution of the types of these disagreements in the evaluations of both systematic and lazy strategies, illustrates the fact that 200 experimental runs are enough to explore the space of random setups. The conclusions that we draw from the fact that there are no final disagreements, of any type, are the same as for the evaluation of the generality of the systematic approach: our model can correctly address and resolve any combination of disagreement types and resolve all of them.

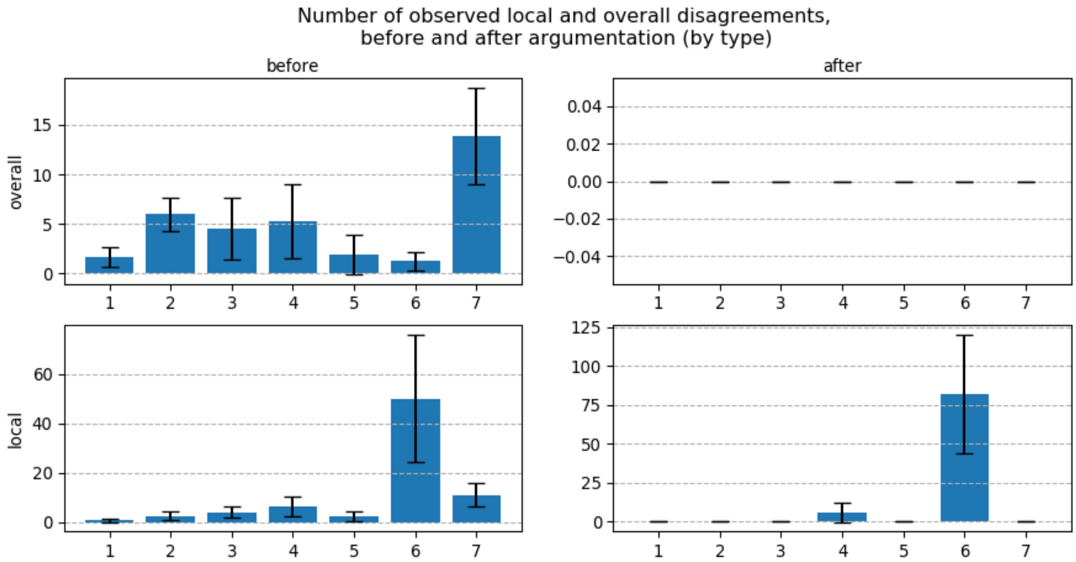


Figure 10.6: Disagreement count in a lazy argumentation with random disagreements. The y-axis displays the count of each of the types of disagreement displayed in the x-axis: 1.Self-Disagreements, 2.Overlaps, 3. Hypo/hypernymies, 4. Synonymies, 5.Homonymies, 6.Indistinguishable disagreements, 7.Untranslatable disagreements

Figure 10.6 shows the count of each type of disagreement before and after the argumentation. The upper plots display the overall disagreements, while the bottom plots display the local disagreements on a individual agent. Here again, we observe results similar to the ones presented in Figure 10.3 which gives us an additional confirmation that similar setups are found in both experiments. The local disagreements found after the argumentation are as well synonymies and indistinguishable disagreements, in similar proportions as in the systematic strategy, which means that the explanation given during the analysis of the systematic strategy results holds here as well.

Figure 10.7 presents the ratio that have been measured, before and after the argumentation. The leftmost plot presents the overall measures. The DAR, that is in essence a local measure which compares two contrast sets of a same agent, is here presented as the average of two DARs of the agents, averaged again over the 200 experiments. The rightmost plot presents the average difference between the local measures of a same experiment. We can observe a noticeable difference between the SAR of the systematic and the lazy strategy, the SAR being on average more than 0.8 in the systematic strategy while also being less than 0.8 in average in the lazy strategy. The DAR is not affected by the argumentation in the lazy strategy, as it was the case for the systematic strategy. However, the exchange ratio is lower in the lazy strategy than it is in the systematic strategy, which gives an explanation to the higher coverage ratio at the end of the systematic strategy. This difference in exchange ratio is explained by the different approaches to the naming game in both strategies. In the systematic approach, the argumentation takes place before the naming game. Since the setup has a 0% redundancy, the agents do not know any example of each other's context when they start to resolve their first disagreement. On the contrary, the lazy approach can see the first argumentation on meaning take place after the presentation of several examples, allowing the agents to have a certain knowledge of the overall context before the argumentation. However, we can observe that this additional knowledge does not translate in a drastic decrease of the exchange ratio, that was already low in the systematic strategy. Finally, the difference between the local measures on the two agents ranges again on a tenth of the global measures. From this, we can conclude that there is again no asymmetry between the agents after the argumentation and thus the symmetry that existed before is preserved.

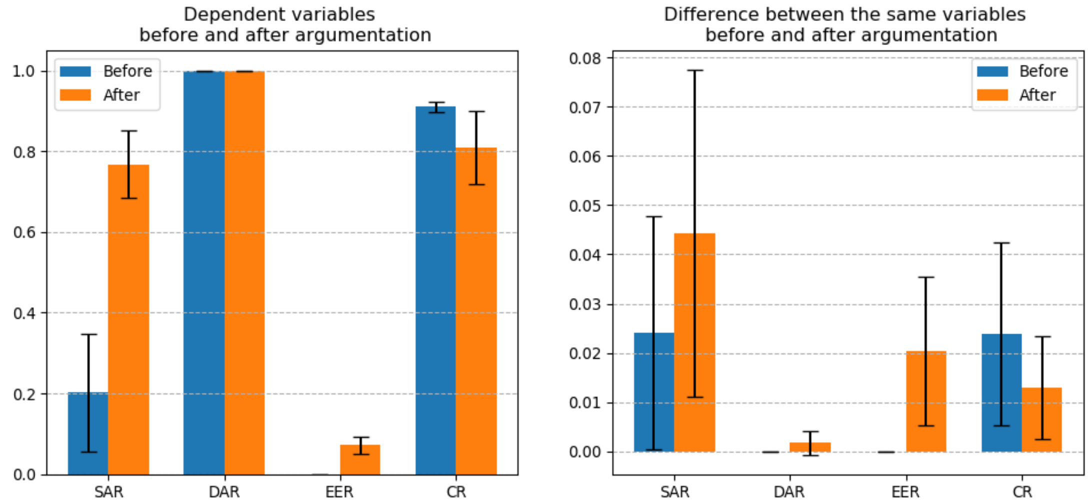


Figure 10.7: Evolution of SAR, DAR, EER and CR before and after a lazy argumentation where the disagreements are randomly selected (left). A local version (taking only the examples of local contexts) of these five variables is implemented and the the euclidean distance between the two agent's variable is measured (right).

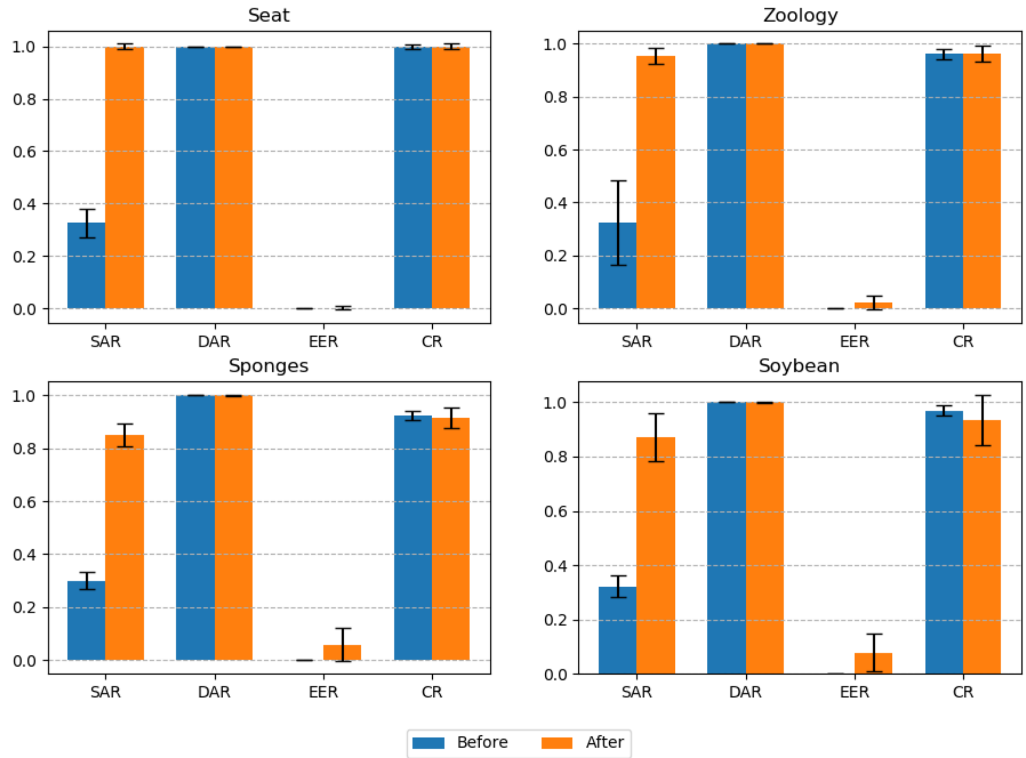
10.3.3 Conclusion of the Generality Hypothesis

Both the systematic and lazy strategies display an increase in their average synchronic agreement while keeping a high diachronic agreement and coverage, and without exchanging many examples. This means that both strategies satisfy the generality hypothesis.

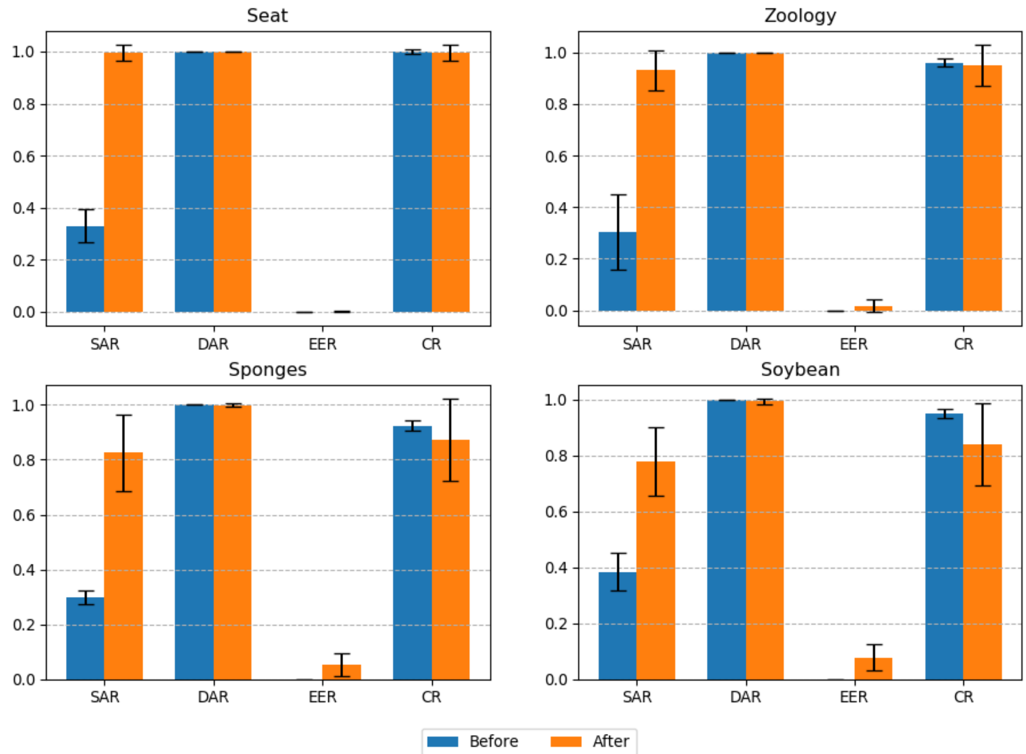
10.4 Domain Independence

The domain independence hypothesis postulates that both approaches of our model can resolve a disagreement between two agents that occurs on any domain, increasing the SAR of the agents while keeping their respective DAR at one. In order to test this hypothesis, we used with four different domains, and set up exactly *one* overlap disagreement between two agents on each of them. Moreover, the examples that have not been used in the set up of the overlap are removed from the data-set of the agents before the beginning of the argumentation, in order to ensure that we do not have more concepts that are not used in a set up in some domains than in others. We tested this experiment 100 times on and averaged the measures in order to present the results below.

In each of the 100 experiments, the SAR, DAR, EEC and CR of the agents are measured in each of the four argumentation occurring on the different domains. The four domains are the independent variables our our experiment, while the SAR, DAR, EEC and CR are the dependent variables. The argument acceptability is 0.75, its standard value, and the redundancy is set at 0% as usual. However, the error threshold is different for each data-set. The error threshold is set in order to ensure that at least 3 classes from a set of classes of comparable sizes are available to set up the overlap in each data-set. We decided on $\tau_E = 1$ for the Seat domain, $\tau_E = 6$ for the Zoology domain, $\tau_E = 10$ for the Sponges domain and $\tau_E = 11$ on the Soybean domain.



(a) Ratios measured before and after a systematic argumentation, over the four different domains *Seat*, *Zoology*, *Sponges* and *Soybean*.



(b) Ratios measured before and after a lazy argumentation, over the four different domains *Seat*, *Zoology*, *Sponges* and *Soybean*.

10.4.1 Systematic Strategy

The results of this experiment are shown in Figure 10.8a. Each of the four sub-figures represents the four ratios measured on a specific domain, before and after the argumentation. We can observe a generalized increase of the SAR for all domains. The SAR before argumentation is on average at a predictable 0.33 for all domains, with a low variability except for the Zoology domain which is the domain where the distribution of the examples among the classes over $2 \times \tau_E$ elements is the most irregular. The average SAR after argumentation can vary from 0.83 to 1.0, depending on the domain. The systematic strategy achieves the highest increase of the SAR, with a final SAR of 1.0, meaning that the synchronic agreement is fully reached. The systematic strategy obtains a final SAR of 0.95 on average on the Zoology domain, and around 0.85 on the Sponges (0.85) and Soybean (0.87) domains. These results correlate with the complexity, in terms of different attributes, of the different domains. The DAR is maintained at 1 in every domain, with a standard deviation always close to 0. There is always a full diachronic agreement for both agents.

There are logically no example exchanged before the argumentation. The augmentations on the Seat domain are done without example exchanges, as the EER stays at 0 with a low standard deviation. On the other domains, the EER increases with the complexity of the domain expressed in terms of different attributes. Zoology accounts for the lowest non-zero EER, followed by Sponges and finally Soybean. The EER evolution is evaluated more in detail with the test of the preservation hypothesis.

The CRs before argumentation correlates on the complexity of the domain. The Seat domain has a CR at 1 with low standard deviation before the argumentation, the Zoology domain has a CR at 0.96, the Sponges has a CR at 0.92 and the Soybean has a CR at 0.97. The CR stays constant for the Seat and the Zoology domain. However, it decreases significantly for the Sponges add the Soybean domains, the decrease being more pronounced in the case of the Soybean domain. The decreases are again correlated to the complexity of the domains, however the final CR always stay close to the value it had before the argumentation. Even after the argumentation, every CR stays above 0.9, leaving the agents with less than 10% of their overall context unclassified on average.

Overall, the performances of our model are high for every domain. The SAR is the most impacted measure by increments in the domain's complexity, with a lowest final SAR at 0.85 in the case of the Sponges domain and a highest final SAR at 1. The average final SAR being at 0.87 for the Soybean domain in this experiment, while it was at 0.83 in the last experiment, leads us to think that the average SAR measured in a situation of set up overlap is a good indicator of the SAR that should be expected for a combination of set up disagreements. The DAR remains at 1, proving once again to be the most stable measure.

10.4.2 Lazy Strategy

The results of this experiment are shown in Figure 10.8b. Each of the four sub-figures represents the four ratios measured on a specific domain, before and after the argumentation. On the contrary of the experiment on generality, the SARs of the two approaches are now comparable in each domain. Since the experiment on domain independence involves one overlap disagreement per experimental run while the experiments on generality involved multiple disagreements of multiple types in each experimental run, this is indicative that the difficulties of the lazy approach to reach the level of synchronic agreement of the systematic strategy, comes from the diversity and simultaneity of the disagreements in the generality evaluation. The DAR is maintained at 1 in every domain, with a standard deviation always close to 0. There is always a full diachronic agreement for both agents. This does not change from the evaluation of the systematic strategy. The average EER and the average CR are comparable to the average EER and CR obtained with the systematic approach in each domain, with a more important standard deviation however in the case of the the CR. The EER and the CR continues to correlate with the complexity of the domain. Overall, the performance of our approach remains high for every domain and the same conclusions as before can be drawn for the evaluation of the domain independence hypothesis now for the lazy strategy.

10.4.3 Conclusion of the Domain Independence Hypothesis

Both the systematic and lazy strategies display on average a net increase of their overall measure of synchronic agreement, while not displaying a decrease in the measure of diachronic agreement, for all domains. Moreover, the coverage of the overall context is preserved for all domains and the proportion of examples from the overall context exchanged by the agents stays low for every domain. Good performance on all domains, regardless of the strategy used, means that both strategies satisfy the domain independence hypothesis.

10.5 Coverage Preservation

The preservation hypothesis postulates that most of the examples that were covered by a concept will continue to be covered by some concept in the refined contrast set. While a diachronic agreement ensures that no pair of examples from the same extensional definition *of the new context* was separated in the initial contrast set, the diachronic agreement does not constraint the new context to be a super-set of the initial context. Therefore, as our model often refines concepts, we can expect that a context after an argumentation is a subset of itself before the argumentation. In this sense, the hypothesis of preservation is complementary with respect to the diachronic agreement in making sure that, after an argumentation, an agent can make *over the overall context* classifications that are equivalent to the ones that it did before classifications.

The preservation hypothesis has already been partially explored during the experiments over the Generality and Domain Independence hypotheses. However, the measures were approximated. Indeed, the overall CR computed was each time obtained as the average between both local CRs. This approximation is only accurate if one local set of covered examples is a subset of the other. In this section, the preservation hypothesis will be specifically investigated by searching for the subset of examples from the overall context that are not covered by *both* agents. In order to evaluate this hypothesis over different contexts, we repeated this experiment 100 times on each domain over which we observed a final CR value lower than 1 during the test of the domain independence hypothesis: Zoology, Sponges and Soybean.

The independent variables of each experiment and their overall setup are similar to the independent variables and setup of the domain independence hypothesis test. One experiment run is set up using three random classes of the selected domain. The domain is the only independent variable. The parameters are the same as for the domain independence hypothesis test. The dependent variables are however different: the overall CR value is observed before and after the argumentation —this time measuring the examples covered by an intensional definition from *both* agents, and is presented with the difference between each local CR value, this time again measured before and after the argumentation.

10.5.1 Systematic Strategy

The results of this experiment are shown in Figure 10.9 (left). The leftmost bars represent the average coverage before the argumentation, while the rightmost bars represent the average coverage after the argumentation. As in precedent figures, we also present the average difference between the average coverage of the agents A_1 and A_2. As we can see, the coverage increases after the argumentation, unlike what we observed in the test of the domain independence hypothesis. This can be explained by the fact that while each agent initially covers most examples, the examples

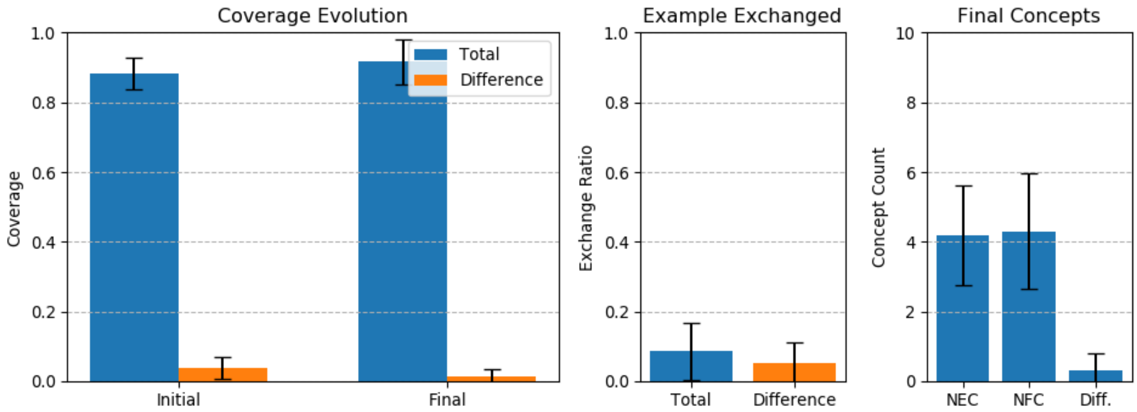


Figure 10.9: Measures of Coverage Ratio (left), Examples Exchanged Ratio (middle) and final Concept Count (right) after a systematic argumentation.

that initially stay uncovered are different from one agent to another. In these situations, the overall disagreements are different from the local ones, and we already know that these situations appear as the setup disagreement are different from the observed ones. At the end of the argumentation, the agents have more concepts that share their intensional definitions, and therefore the agents took advantage of each other's coverage. This is reflected in the difference between the final covering of our two agents, which is a smaller difference than the difference between the initial covering of our two agents.

10.5.2 Lazy Strategy

The results of this experiment are shown in Figure 10.10 (left), with the same template as in Figure 10.9. We can observe that the coverage ratios are similar in both strategies. In conjunction with the results of the experiment on generality and domain independence hypotheses, this illustrates that the lazy strategy, as the systematic strategy, reduces the proportion of examples covered by either agents' contrast set but increases the proportion of examples covered by both.

10.5.3 Conclusion of the Coverage Preservation Hypothesis

Both the systematic and lazy strategies display an increase in the average proportion examples covered by the contrast sets of both agents after argumentation. This means that both strategies satisfy the coverage preservation hypothesis.

10.6 Efficiency

In order to reach an agreement, the agents could transfer their entire contexts to each other. The issues of finding the overall pairing relations from the local ones and of creating new concepts through argumentation would disappear. The remaining tasks for our agents would only be to define new extensional definitions, learn satisfying intensional definitions for them and elect a new lexicon for the resulting contrast set. An important part of our model is dedicated to circumvent the problems arising when we want to avoid transferring *all* the examples. The efficiency hypothesis proposes that the mutual intelligibility may be reached by the agents without exchanging their contexts so extensively. In order to test this hypothesis over different contexts, we repeated this experiment 100 times on each domain over which we observed a non-zero EER value during the test of the domain independence hypothesis, namely Zoology, Sponges and Soybean domains.

The independent variables of this experiment, its overall set up and its parameters are the same as for the test of the domain independence hypothesis. The dependent variables measured are the final overall EER value and the difference between the two final local EER values.

10.6.1 Systematic Strategy

The results of this experiment are shown in Figure 10.9 (middle). The leftmost bar represents the average total example exchange ratio, while the right bar represents the average difference between the EER of A_1 and A_2. As we can observe, the difference between the EER of A_1 and A_2 is close to the total EER which means that one agent is often in charge of sending the examples to the other. This is in fact due to the nature of the experiment: as we only setup one overlap, there is only one new intensional definition that needs to be created (as the intensional definitions of the two co-hyponyms created during the resolution of the two hypo/hypernymy disagreements can be reused from already existing concepts). Therefore, only one agent needs to send examples to the other (possibly during the creation of the new intensional definition, but most likely during the computation of the overall definition). Figure 10.4 was displaying a smaller difference, which tends to confirm that hypothesis: if one agent is mostly in charge of the examples sent for the resolution of a disagreement, the fact that we are setting up more disagreements allows both agents to send examples and levels the difference.

The example exchange ratio itself is low compared to the gain in term of synchronic agreement. In Figure 10.8a, we can see that the gain of SAR after the argumentation is always more than 0.4. An example exchange ratio of lesser than 0.1 confirms that the gain of SAR is not only due to the gained similarity of the two agents' contexts.

10.6.2 Lazy Strategy

The results of this experiment are shown in Figure 10.10 (middle). The leftmost bar represents the average total example exchange ratio, while the right bar represents the average difference between the EER of A_1 and A_2. We can observe that less examples are exchanged during an argumentation using the lazy strategy, observation that goes in the direction of other observations in the test of the generality and domain independence hypothesis. As for the systematic strategy, the difference in examples sent between the two agents is on average comparable to the number of examples exchanged by the agents, meaning that an agent is often sending all the examples exchanged to the other agent during an experimental run. This fact is again explained by the fact that this experiment sets up a unique disagreement, and therefore only one disagreement needs to be identified which is therefore done by only one agent.

10.6.3 Conclusion of the Efficiency Experiment

Both the systematic and lazy strategies display a small portion of the overall context of the agents exchanged during the argumentation. This means that both strategies satisfy the efficiency hypothesis. Moreover, we observe that more examples exchanged, on average, with the systematic strategy than with the lazy strategy.

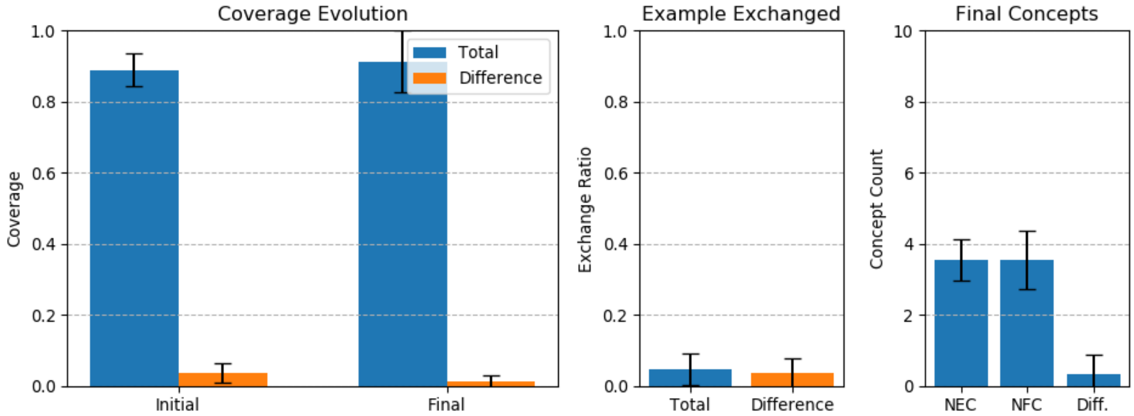


Figure 10.10: Measures of Coverage Ratio (left), Examples Exchanged Ratio (middle) and final Concept Count (right) after a lazy argumentation.

10.7 Simplicity

In Chapter 3, we explained the difficulty to differentiate between overlap disagreements which should lead to the creation of a new concept, and small overlaps caused by the agent's underfitting initial learning. An underfitting initial learning creates concepts that are subsuming more examples than they should in the overall context, resulting in overlap or self-disagreements when two unrelated concepts are subsuming parts of each others examples. However, while setting up the experiments, the number of concepts in the initial data-set is known with certainty. Comparing this initial number with the final number of concepts present in the contrast sets *after* the argumentation on meaning allows us to see how many additional concepts have been created during the argumentation. However, this measure would be imperfect in the sense that it would penalize our model for something it is not directly aiming to solve: the accuracy of the agent's initial learning.

As the argumentation on meaning takes place after the initial learning, the measure of simplicity should take into account *all* the overlaps between the two contrast sets' concepts, which corresponds to the polylexematic classes of the union of the two agents' left-paths associations. By counting these polylexematic classes that contain more examples than the error threshold, we obtain the number of expected concepts (NEC). Comparing the NEC with the number of final concepts (NFC) gives us a measure of the simplicity of our model.

Our hypothesis of simplicity is that the average difference between the NEC and the NFC is close to 0. A difference significantly above 0 would mean that our model converges by creating more concepts than necessary, which should increase the costs of our argumentation model in terms of example and generalization exchanges. A difference significantly below 0 would mean that our model creates less concepts than necessary, which should decrease either the synchronic agreement ratio or the coverage ratio. In order to evaluate this hypothesis over different contexts, we repeated the Domain Independence experiment 100 times on each domain over which we observed a final CR inferior to 1 during the test of the domain independence hypothesis: Zoology, Sponges and Soybean. Each time, we calculated the NEC and the NFC, subtracting the former to the latter.

10.7.1 Systematic Strategy

The results of this experiment are shown in Figure 10.9 (right). The leftmost bar represents the average number of expected concept while the middle bar represents the average number of observed concept. These two values being close to each other is not enough to conclude that the final number of concepts is the expected number of concept. Indeed, the NEC and NFC being

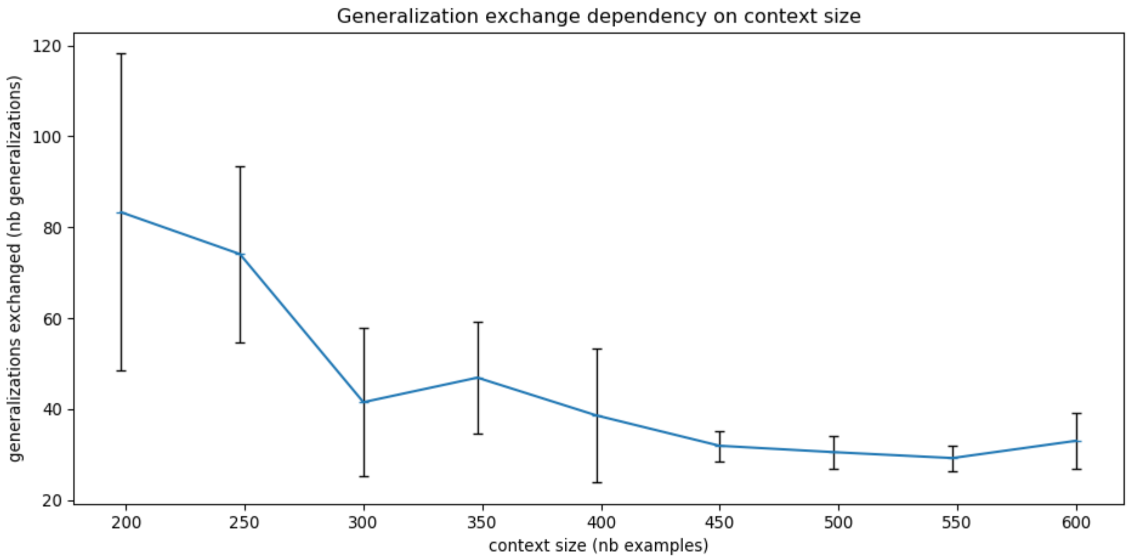


Figure 10.11: Variations of the number of generalizations exchanged during a systematic argumentation according to the size of the overall context.

averaged over a hundred of experiments does not prove that in a same experiment, the two values are always close. However, the rightmost bar represents the average distance between the NEC and the NFC. The euclidean distance between the NEC and the NFC is calculated for each experiment and then averaged. The low result, lesser that one, shows that on average the NEC and NFC are the same, and that our model is correctly replicating the complexity that could have been obtained by a brute-force approach. Our model is therefore not creating additional concepts that are not necessary to reach mutual intelligibility.

10.7.2 Lazy Strategy

The results of this experiment are shown in Figure 10.10 (right). The leftmost bar represents the average number of expected concept while the middle bar represents the average number of observed concept. The values are here lower than for the experiment on the systematic strategy, which just means that the agents are on average initially better at learning concepts close from the three expected ones in their initial overlap setups. This does not affect in any way the results of the experiment. However, the observation that the average NEC and NFC are really close to each other, conjugated with the observation that the average difference between the NEC and the NFC in a run is close to 0, allows us to draw the same conclusion as for the same experiment on systematic strategy: the agents are not creating more concepts than needed.

10.7.3 Conclusion of the Simplicity Experiment

Both the systematic and lazy strategies display on average the same number of expected and observed concepts, with almost no difference between numbers of expected and observed concepts in each experimental run. This means that the simplicity hypothesis is validated for both strategies.

10.8 Scalability

The efficiency hypothesis questioned the ability of our model to reach mutual intelligibility while exchanging a reasonable amount of examples. The scalability hypothesis is the same notion applied to the exchange of generalizations. While we expect our model to exchange a number of generalizations significantly higher than the number of examples exchanged, the scalability hypothesis investigates the correlation between the number of generalizations exchanged and the size of the overall context. Our hypothesis is that if the number of generalizations exchanged increases polynomially with the size of the overall context, then our model should be computationally efficient once the size of the overall context reaches a certain threshold.

In order to test this hypothesis, we created a set of 10 artificial data-sets that have larger classes than the data-sets used in the other hypotheses evaluations. All the data-sets have 600 examples divided in three classes of 200 examples each, allowing us to test the cost of our argumentation protocol for different sizes of context. Each time, we set up an overlap disagreement with the three classes. The examples from the artificial data-sets are a vector of a attributes-value pairs. Each attribute can take a value from a set of l values. The artificial data-sets are generated with predefined rules. The rules are three sets of m subsets of n attribute-value pairs. For each rule r, there is exactly one hundred examples that share n attribute-value pairs with r. Each of these examples only shares n attribute-value pairs with r. Each data-set has a different combination of parameters l, m and n:

- l varies from 6 to 10,
- m varies from 1 to 4,
- n varies from 2 to 6.

There is therefore a total of 100 different artificial data-sets, each having 600 examples. The scalability hypothesis is tested by measuring first the effect of context size on the number of generalizations exchanged between the agents, testing how many generalizations are exchanged in our model on an average of ten runs for each size of context ranging from 200 to 600 examples, with an increment of 50 examples. The artificial data-set used for the test uses the parameters $l = 6$, $m = 4$, $n = 2$. Moreover, in order to observe how the different parameters influence the number of generalizations exchanged, we measured in a second time the average number of generalizations exchanged in our protocol over ten runs for each possible combination of the parameters l, m and n. In this second part of our experiment, only the first 200 examples of the artificial data-sets are kept as the overall context of our agents.

10.8.1 Systematic Strategy

Figure 10.11 represents the average number of generalizations exchanged for each size of overall context ranging from 200 to 600. We can observe that there is no positive correlation between the number of examples in the overall contrast set and the number of generalizations exchanged by the agents. Since the rules that were used to build the contexts, and that should correspond to the final intensional definitions of the agents' contrast sets after reaching mutual intelligibility, are the same for each context size, it was expected that the generalizations exchanged are the same and should not have an impact on our result. Moreover, an increased number of examples in the contexts means that the agents have more examples from which to learn their generalizations. Therefore, the agents are expected to classify more accurately during the creation of their concepts and their counter arguments. This is reflected by a progressive drop of the number of generalizations exchanged past 150 examples in the overall contrast set, going along with a decreasing variability

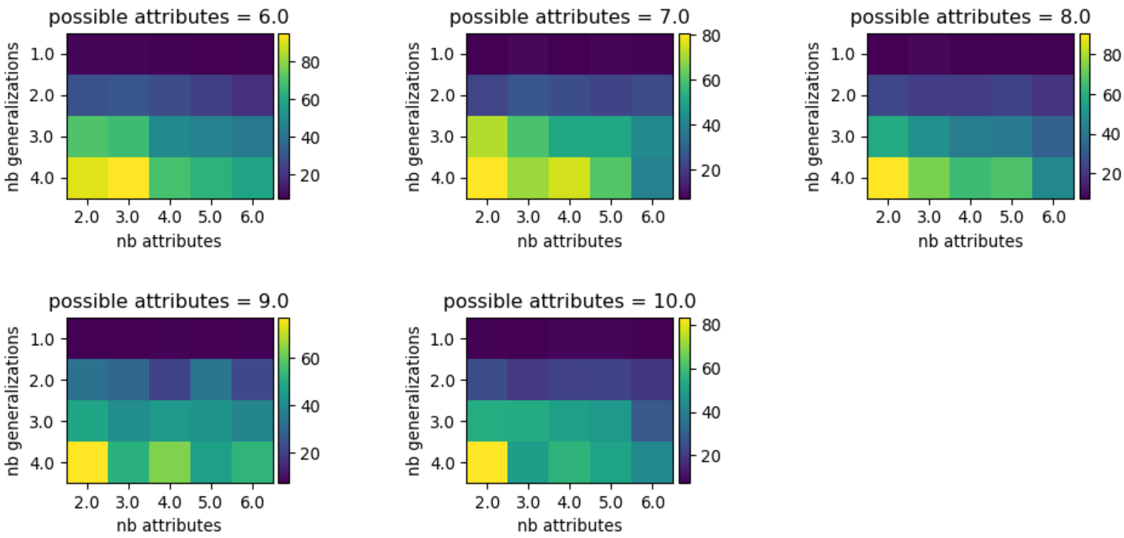


Figure 10.12: Variations of the number of generalizations exchanged (color) according to the parameters of the data-set.

in the result: the inductive learning being more accurate, the chances of having disagreements that were not set up, and the chances of having arguments with false positives or negatives, decrease.

The results presented in Figure 10.11 are obtained for a fixed set of parameters ($l = 6$, $m = 4$, $n = 2$). We can observe in Figure 10.12 that this set of parameters is in fact the one of the configurations for which the agents exchanged the most generalizations. The number of generalizations exchanged tends to increase with the size of the expected intensional definitions, and to decrease with the number of possible values for each attribute and the number of attributes in each of the generalizations from the expected intensional definitions. According to these results, the configuration of the data seems to impact the argumentation more than the number of examples involved in the argumentation.

10.8.2 Lazy Strategy

Figure 10.13 represents the average number of generalizations exchanged for each size of overall context ranging from 200 to 600. We can observe that there is no positive correlation between the number of examples in the overall contrast set and the number of generalizations exchanged by the agents. We can observe that the average number of generalizations exchanged on contexts having 200 examples is higher than for the systematic strategy, but with a variance that makes this result inconclusive. However, we can observe that the plot shown in Figure 10.13 has the same shape as the plot shown in Figure 10.11, the average number of generalizations exchanged and their standard deviation being inversely proportional to the size of the context. We can also observe that for a size of context over 400 every argumentation run exchanges an average of around 40 generalizations. Overall, we observe that on average an argumentation using the lazy strategy performs more exchanges of generalizations than an argumentation using the systematic strategy with an equally sized overall context. This is explained by the fact that agents in the lazy strategy are creating new concepts in the restricted context of their currently investigated connected sets of disagreements. While the agents can create satisfying meanings for their new concepts within the limit of their current connected set of disagreements, errors in generalization during the concept creation —even errors within the limits of our the assumed degree of error τ— can result in the new concept causing disagreements with other concepts that are not included in the set of disagreements. The late inclusion of these concepts in the set of connected disagreements can result in an order of

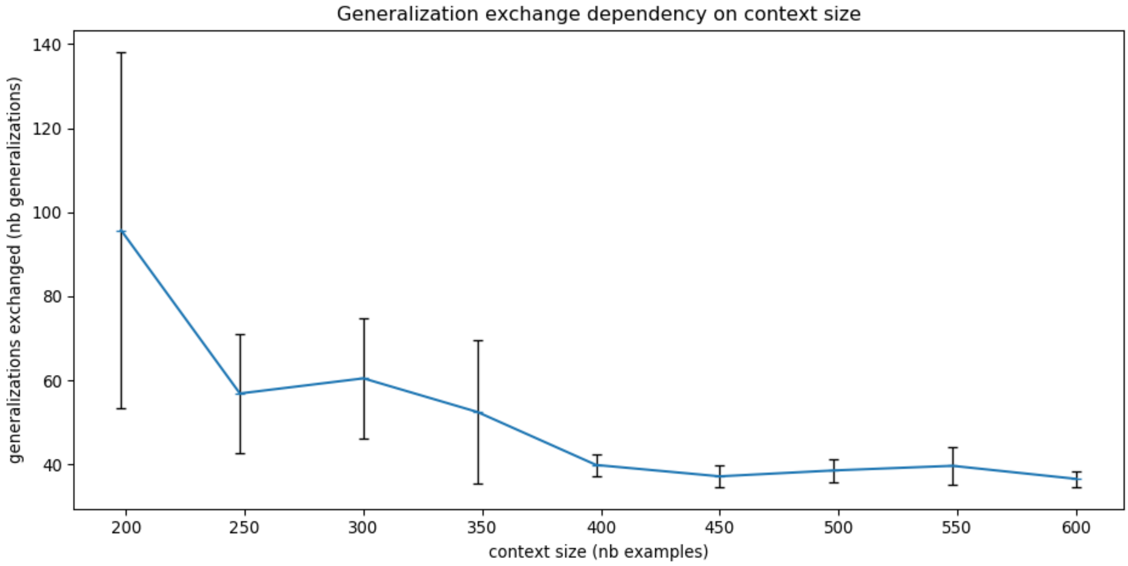


Figure 10.13: Variations of the number of generalizations exchanged during a lazy argumentation according to the size of the overall context.

resolution of the disagreements that is different from the optimal order presented in Section 5.5.6, which is the order that is always followed in the systematic strategy. This means that the agents will go through more concept creations before resolving all disagreements.

10.8.3 Conclusion of the Scalability Experiment

Both the systematic and lazy strategies do not display any polynomial increase of their number of generalizations exchanged when the size of their context increases. This means that both strategies satisfy the scalability hypothesis. Moreover, we observe more generalizations exchanged on average with the lazy strategy than with the systematic strategy.

10.9 Conclusion

Each of our six hypotheses have been experimentally tested and validated with both strategies. Our model, overall, is general in terms of disagreements resolution, independent of its domain of application, preserving the coverage of the overall context by both agents, efficient in example exchanges, simple in the set of concepts that it yields, and the number of generalizations that it exchanges scales well with the size of its domain of application. The systematic and the lazy strategies, while being different in their approaches to the naming game, do not produce greatly different results, except for the type of elements exchanged with the highest count. We observed that the systematic strategy privileges the exchange of examples, as the lack of interaction between the agents before the resolution of disagreements forces them to exchange some parts of their context that the lazy strategy would already have explored during the early turns of the naming game, while the lazy strategy privileges the exchange of generalizations as it does not research all the pairing relations between every concept at once and therefore can go through the sequential resolution of disagreements in a less optimal order than the systematic strategy.

Chapter 11

Conclusions

11.1 Contributions

The contributions of this thesis concern the innovations that we introduced in order to integrate three Artificial Intelligence fields that are, in practice, distant from one another: 1) symbolic concept learning; 2) coordination and argumentation in agent systems; and 3) semantic alignment.

Concerning Symbolic Concept Learning

Concerning symbolic concept learning, we keep the assumption that all examples are represented in a common representation formalism and a common vocabulary —what is classically known as the representation bias in ML systems. However we change two important issues: how labels apply to examples, and dealing with collections of concepts in learning as a whole.

In symbolic concept learning, labels apply to classes as a simple assignment relation that is fixed, and cannot be changed or enlarged. In our approach, labels are conceptualised as signs included in a semantic triangle model of concepts, and this is integrated in the concept learning process. The assignment becomes an association of signs to examples and generalizations, constituting the semiotic triangle, and different semiotic triangles associating different generalization sets (intensional definitions), different example sets (extensional definitions), and different signs are possible. Our approach, in fact, is based on this multiplicity of concepts to explore possible transformations of the initial concepts of two agents, in order to reach a common agreement over those concepts. These transformations include the creation of new concepts.

Concerning the notion of contrast sets, the introduction of this notion allows us to deal with collections of concepts in learning as a whole. Recall that *symbolic concept learning* in ML addresses only learning one individual concept. In a standard classification task, concept learning is applied individually to each class to learn its "concept" (set of generalizations). This standard approach considers examples in other classes as simple counter-examples —regardless of which class they belong to. Our approach, however, considers and addresses directly the fact that the class to which a such counter-example belongs (in the contrast set) is highly relevant. We need to consider the importance of counter-example classes, as classes and concepts can be changed from the initial setup. Therefore, by using a contrast set, with concepts represented in their triangular form during both the learning process and the argumentation process, our approach deals with collections of concepts in learning as a whole. In our approach, what is usually called "contextual meaning" is modelled by the relation of a concept with its "neighbouring concepts" that are relevant together *because* they constitute a contrast set, which is a segmentation of the world that is meaningful for a particular situation or task.

Thus, the semiotic view of concepts via the triangle model (with the distinction of left and and right path) and the characterisation of classification tasks as contrast sets that meaningfully segregate the world and categorize its elements, are both contributions to the general area of symbolic concept learning.

Concerning Argumentation Between Two Agents

Concerning argumentation between two agents in multiagent systems (MAS), we want first to remark that argumentation is classically conceived as taking place between two individuals — including human individuals. When a larger number of individuals are involved, such as in parliaments or commissions or committees, they need to be addressed as *deliberative bodies* that require institutional norms and procedures to achieve collective actions and agreements. When a number of individuals is small, or at least the number of other individuals with which one individual interacts is small, it can be practical to model their interaction as a collection of two-agent argumentation processes.

Our approach combines agents that learn and agents that argue, and it builds upon this combination to generalise the multiagent learning approach proposed in the AMAIL framework (Ontañón and Plaza, 2015). However, the goal of the AMAIL framework was only to provide learning capabilities to agent systems, and as such the restrictions and assumptions of classical symbolic learning were maintained. Our approach here is capable of performing argumentation about classification tasks where the two agents have different number of classes and/or different names (signs) for their classes, which was impossible in AMAIL. The new representation model of concepts — the semiotic triangle— allows our model to deal with the differences in number of classes, sign associations, and intensional definition covering, including the creation of new classes and the negotiation of an agreement on classes, signs, and their associations. In contrast, AMAIL was only capable of argue about the details of the shape of borders by modifying intensional definitions of a pre-established set of concepts. Moreover, a typology of disagreements and a method to derive an overall model of disagreements from individual agents' viewpoint on disagreements is what provides our approach with the knowledge to decide the individual actions of the two agents in the process of resolving disagreements over meaning and reaching mutual intelligibility.

Finally, notice there is an argumentation/learning continuum that is covered in our approach. Although the strategy we present uses argumentation to decrease the amount of information exchanged, our model also covers scenarios that could be conceptualised as mostly learning. For instance teacher/apprentice scenarios, where agents are asymmetrical, such that a teacher agent has much more examples tan the apprentice, the "correct" number of classes, and the correct signs naming them. Such scenario is also amenable to be covered in our model, since the outcome would be that one agent provides a lot of examples unknown to the apprentice agent, and the apprentice would have less content from which to extract examples or generate arguments. Nevertheless, the apprentice also being able to argue would mean that the information coming from the teacher would not be more than what needed to reach a state of mutual intelligibility, instead of just sending all the examples from the teacher to the apprentice. The experiments we used in our evaluation were however all symmetrical, because they represent the set up that is the most complex for argumentation while we wanted to evaluate our approach in the most exacting circumstances.

Concerning Semantic Alignment and Evolution

Coordination —and even communication— in multiagent systems often assume that the agents share an ontology, or at least some kind of knowledge that kind be represented under an ontological form. In this view, before a multiagent system (MAS) can interact, it is assumed there is a previous step by which a shared ontology is defined by some authority, or is achieved by some means: by ontology engineering using ontology alignment or mapping, for instance. In other words, this problem is of a purely technical nature, and can be solved by an independent discipline outside MAS. This assumption, of course, can have practical value for certain MAS applications. However, this assumption is too restrictive for open multiagent systems since it assumes no misunderstanding/disagreement on meaning need to be addressed during the actual operation of a multiagent system.

Our viewpoint is that this "fixed meaning" precondition can be too strong in certain scenarios, and we show that this is not always a necessary precondition, as there are ways to deal with disagreements/misunderstandings and to reach agreements —albeit contextually bounded to certain agents cooperating in a shared environment. We call our approach mutual intelligibility, for this reason: we do not see this as a technical problem that has an engineering solution —ontology alignment is defined as a process to help "interoperability" of systems—, but as a a problem that involves semiotics, meaning how and when a sign is associated to an entity as being a member of a concept or category, and a contextual view of meaning, modelled by the notion of contrast set. In this approach, misunderstandings and failures in communication are viewed as disagreements over meaning — and we have characterised how differences in the associations used to relate signs to elements, or signs to intensional definitions can be identified and disagreements can be resolved.

Certainly, a necessary property of our agents is that they are capable of learning: our approach would not work on agents that are programmed as a fixed algorithm to decide their actions —such agents indeed need an ontology engineering process that insures no communication failure can occur. This final consideration remarks how crucial is the integration of the three AI fields not apply only to our approach, but to the general problem of MAS concerning open systems, semantic alignment and evolution, capability to adapt to new agents or new vocabulary, etc.

Specific contributions

In addition to this general contributions, there are specific and novel ideas, definitions, models and methods that are presented in the thesis and that are needed for sustaining the general contributions.

In presentation order, the first specific contribution of our approach was to find a use to the semiotic triangle to define the notions of *left-path and right-path associations.* These two notions that we are introducing are useful to explicit the preference of a system for one type of association in a particular paradigm and to integrate the learning capabilities and argumentation capabilities of agents from a semiotic perspective.

We then proposed the notion of *adjunct sets* of concepts to represent the contextual meaning of a concept in another context than the one in which it has been learned. Doing so, we were able to define a set of *pairing relations* for concepts, that connect pairs of concepts with a new *typology of disagreements*. Finally, we proposed a strategy for agents to figure out what the relations between any pair of concepts is in the agents' overall context, without having to have direct knowledge about these examples.

The contributions introduced in the Approach chapter are used to design our computational model of argumentation. This new model contributes to the field of multiagent systems with the development of two strategies of argumentation that are error tolerant and able to resolve any type or combination of types of disagreements, increasing the level of mutual intelligibility between two agents. The efficacy of both strategies in resolving all and any disagreements (and their combinations thereof) is a main contribution for runtime autonomy and capabilities for agents addressing semantic heterogeneity.

The remaining contributions of our argumentation model correspond to satisfying the six hypotheses we presented as valuable. According to the results of our experiments, our model is:

1. Assessing and validating the identification and resolution of all disagreements, from all types and in every domain, while using any argumentation strategy (efficacy).
2. Assessing and validating relative efficiency of our approach on several dimensions
3. Assessing and validating that our approach is successful in reaching mutual intelligibility for all the disagreement types we have identified (generality)
4. Assessing and validating that our approach is successful in reaching mutual intelligibility for a number of multiple combinations of the disagreement types (generality)
5. Assessing and validating that our approach is successful in reaching mutual intelligibility in several domains of different kind and with different bias and properties (domain independence)
6. Assessing and validating the ability of our approach to preserve the context in which the agents are able to interact through the argumentation (coverage preservation)
7. Assessing and validating the relatively low cost of our approach in example exchanges (efficiency)
8. Assessing and validating that our approach does not create more concepts that are not needed in order to reach mutual intelligibility (simplicity)
9. Assessing and validating that our approach can work on large contexts without exchanging more generalizations (scalability).

Scope of contributions

Notice that our approach, in the abstract, is quite general, although some limitations exist due to our two basic assumptions. One limitation is due to our commitment to symbolic learning, so other forms of learning are, in principle, not claimed by us as compatible with this approach. One clear assumption is the *ML representation bias*, which for our research aims means that we assume the representation and terminology used in *example description* is known and shared by both agents. Recall, however, that Machine Leaning has very different techniques but they assume "data" (examples) has a specific, given format —achieving that is called data wrangling in data science. We discuss relaxing this restriction in the next section.

A second limitation is related to the representation formalism. In this respect, our approach requires a very limited assumption: any representation formalism that includes the subsumption relation $\sqsubseteq$ is compatible with our approach. Thus, although our implementation is based on feature terms, representation formalisms used in symbolic machine learning like Horn clauses inductive logic programming, and description logics (where more recently some inductive learning techniques are being developed) are compatible with our approach.

11.2 Future Work

Larger Multiagent Systems

Our model is now applicable to two-agents systems. In future work, we plan to study how our model can be applied to larger systems where each agent has a limited number of agents it interacts with. Doing so requires to present the larger system as a collection of pairs of agents with restricted contexts. In a first time, each pair of the larger system would use our model to perform a specific task that requires both cooperation and contextual mutual intelligibility from the agents in order to be achieved. In a second time, we will present how the different pairs of agents can autonomously separate an overall task into contextual tasks that each pair can help to resolve partially, in order to achieve the overall task in a decentralized manner without requiring an overall consensus on the agents' meanings.

Recursive Argumentation over Concept Attributes

Our model is now applicable to a unique contrast set, meaning that two discuss more concepts the agents need to increase the size of their contrast set and possibly address a geometrically increasing number of disagreements between these concepts. In order to overcome this issue, we will implement the notion of conceptual web in future work. By having concepts in conceptual web, we will not assume shared representations over the properties of examples and intensional definitions anymore but instead represent each of these properties as a concept in its own contrast set. Typically, each property will be understood in an attribute-values perspective, where a certain attribute will be assimilated as a contrast set that has its values for concepts. Doing so would allow us to partially order every contrast set, and to propose a model where two agents can identify for any disagreement which contrast sets are problematic to find a mutual agreement. From then, the agents would be able to resolve disagreements in contrast sets of a lower order first — as their own properties can then be assumed to be shared representations— and continue to higher order, until mutual intelligibility has been reached in for problematic contrast sets.

This model would still require a lower level of properties that are considered as shared representations. Future work will also include developing a model where agents can detect situations when no shared level of properties, and use an extensional-only argumentation protocol to create this layer, in a manner similar to language acquisition.

Appendices

Appendix A

Evaluation of the Parameters

In Chapter 10, we briefly presented three parameters along with the usual independent and dependent variables. These three parameters of our model, redundancy, error threshold and argument acceptability are not tested as variables for two reasons: they are expected to impact the results in a predictable way, and they are two costly to test in terms of computation time to be extensively testes. However, the first of these two reasons is a hypothesis and, while we are not able to test it properly, we can give an overview of the reasons that led us to think that the redundancy, the error threshold and the argument acceptability have expected impacts on the results and which are these impacts. This is why the effects that variations of these parameters have on our model are tested below.

Instead of testing these parameters as variables during different augmentations, which would take a prohibitive amount of time, they are tested as variables over several different data sets in the creation of initial contrast sets. After the creation of the initial contrast sets, we can count the number of examples of each of the 2^{m+n} adjunct sets' intersections $Adj(i, U_O) \cup Adj(j, U_O)$ for each pair of concepts $C_i \in K, C_j \in K'$ with K having m concepts and K' having n concepts. The number of intersections that have at least τ_E examples is the number of concepts that we should expect from a brute-force approach of the argumentation over the meaning.

The number of expected concepts is a good indicator of how the inductive learning of ABUI is going to perform over a given set of data, for a given degree of error and redundancy. The degree of error tolerated gives additional information on the values of τ_E that should be chosen in order to find a final number of concepts close to the number of categories of the data set used in the experiment.

We are testing these three parameters as variables over three different data sets: the zoology data set, the soybean data set and the sponges data set. We test the argument acceptability going from 0 to 1., with an increasing step of 0.05. The redundancy is tested on a range from 0% to 100%, but with an increasing step of 50%. However, we chose to let the number of examples in the initial contexts be the same regardless of the redundancy, for a given data set. In order to do so, the number of examples of each category Ca in each agent's initial context is $|Ca|/2$. For instance, the category *astrophorida* has 40 instances in the Sponges data set. Each agent will receive 20 examples from this category. If the redundancy is 100%, they will receive the same subset of 20 astrophorida examples – which still might be labeled differently. If we had distributed 21 examples, we could have still generated two different subsets of 21 examples that are 100% redundant; however, it would have been impossible to find two disjoint subsets of 21 astrophorida examples that are 0% redundant (there is less than 21×2 astrophorida examples in the Sponges data set).

The value of τ_E, however, is not tested on the same range for all data sets. This is due to the fact that τ_E should be smaller than half of the examples of a certain amount of categories in order to allow the agents to find overall relations between their resulting concepts. For instance, if we set a $\tau_E = 20$ in an experiment that uses a data set where the largest category has 30 examples, each agents will receive a set of $30/2 = 15$ examples of this category. Since $15 < \tau_E$, the pairing partial sets between the agents will always have a cardinal lower than τ_E and the agents will remain blind to the pairing relations between their concepts, thus forbidding an argumentation. For this reason, τ_E ranges from 0 to 5 for the Zoo data set, allowing to use the examples from 4 categories; τ_E ranges from 0 to 10 for the Soybean data set, allowing to use the examples from 6 categories and finally, τ_E ranges from 0 to 15 for the Sponges data set, allowing to use the examples from the 3 categories of the data set. On each of the data set, we test τ_E over the aforementioned range with an increasing step of 1.

A.1 Impact of the parameters over inductive learning

The experimental set-up of each instance of our three parameters' test is realized through the random selection of three different categories in the tested data set, that are merged differently for each agent in order to produce a scenario of expected overlap as it has been already presented in Chapter 9. Therefore, the number of expected concepts that we observe should always be compared to the three initial categories that have been used to set up the experiment. In an ideal case, we should observe three expected concepts: one for each involve category from the data set.

The overlap disagreement has the advantage to also produce two hypo-hypernymy disagreements, allowing us to have an expected initial situation with both types of semantic disagreements. We tested 10 times each combination of redundancy and argument acceptability for each of the three data sets, and presented the average number of expected concepts according to τ_E in Figure A.1. The results show that increasing the redundancy has a high impact on the number of expected concepts, which is easily understandable: by sharing more examples the agents can create generalizations that are as relevant in both contexts, thus explaining why the number of expected concepts converges to three - the number of involved categories from the data set - when the redundancy reaches 100%. The results are presented in Figure A.1

The first thing to observe is that a low percentage of redundancy produces a number of expected concepts closer to three. If the agents start with the same examples in their contexts, they create generalizations that are more likely to correspond to the canonical cases already described in the Chapter 9. While this eases the argumentation, we want to test our argumentation protocol in the worst case of redundancy scenario and therefore, the redundancy parameter will always be set to 100% in our experiments

With a redundancy set at 100%, we observe that the number of expected concepts depends mostly on the degree of error tolerated, regardless of the data set. With a low degree of error tolerated, we obtain too much concepts as the agents learn over-fit intensional definitions for their concepts. We clearly see that this issue peaks with the Sponges data set, where the number of expected concepts peaks over 9 for an error degree of 1. The lowest degree of error that provides consistently a number of expected concepts close to 3 depends on the data set observed: while $\tau_E = 5$ is already satisfying for the Zoology and the Soybean data set, $\tau_E = 8$ is a minimum for the Soybean data set.

Finally, the argument acceptability does not seems to impact the number of expected concepts, at the exception of its highest values. For each combination of data set, redundancy and degree of error, we can observe a constant number of expected concept until the argument acceptability reaches 0.95, where the number of expected concepts drops drastically, often to 0. The Figure A.2 presents the variation of the number of expected concepts with the sponges data set, for a redundancy of 100% and argument acceptability values close to 0.95. We can see that the argument acceptability does not influence the number of expected concepts under the value of 0.94. In our

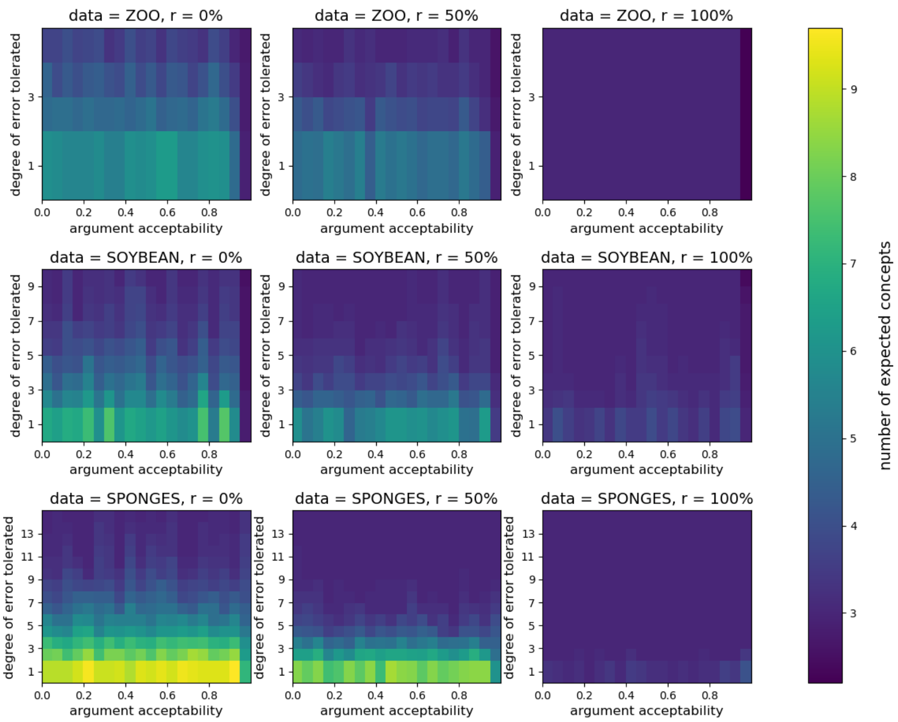


Figure A.1: Number of expected concepts for each combination of degree of error tolerated τ_E, redundancy r and argument acceptability, within the data sets Zoology (ZOO), Soybean (SOYBEAN) and Sponges (SPONGES).

experiment, we chose an argument acceptability of 0.75 commonly encountered in publications using AMAIL, which is below this threshold.

A.2 Impact of the error threshold over argumentation on meaning

After analyzing the different parameters of our experiments, we observed that the parameter which impacts the most the results of inductive learning is the error threshold. In order to evaluate the impact of the error threshold not only on inductive learning, but on a full argumentation, we are setting up a second experiment in which we measure three dependent variables: the initial synchronic agreement, the final synchronic agreement and the number of exchanged examples. The initial and final synchronic agreement are respectively measured at the beginning and the end of an experiment, using the Synchronic Agreement Ratio or SAR. The SAR is detailed in Section 10.1.2, and corresponds to the number of examples from the overall context that are named with the same unique sign by both agents divided by the total number of examples in the overall context. The number of exchanged examples is also measured as a ratio, corresponding to the number of examples that have been sent through messages by both agents divided by the total number of examples in the overall context.

The impact of error threshold is tested on two different data-sets: Sponges and Soybeans. For

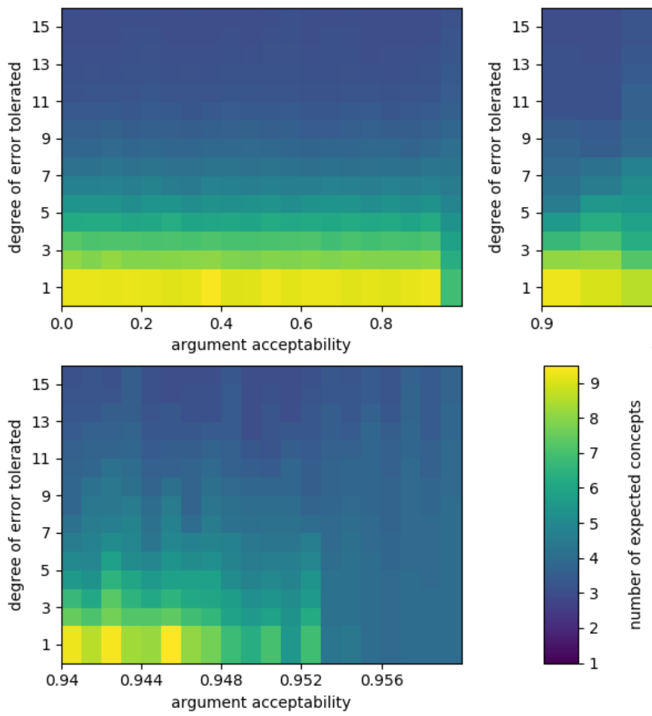


Figure A.2: Impact of argument acceptability on the number of expected concepts on the Sponges data set and for a redundancy of 100% between the two agents' data sets.

each data-set, we set the argument-acceptability to 0.75 and the redundancy to 0%. The error degrees tested vary from one to ten for the Sponges data-set, and from one to fifteen for the Soybean data-set. This difference is based on the comparative sizes of each data-set categories. The argumentation follows a systematic strategy over a setup of as many overlap as the threshold allows.Recall that in order to setup one overlap we need three categories of the data-set, and that in order to use a category of the data-set the number of examples in that category should not be lesser than twice the error threshold. This results in a major difference between the experiments on the two data-sets: while the Sponges data-set is limited to one overlap using any combination of its three 40-examples concepts regardless of the error threshold, the number of overlap increases as the error threshold decreases with the Soybeans data-set since the 19 categories have different numbers of concepts. The systematic strategy and the overlap setup are privileged as they correspond to a baseline setup: the lazy strategy is an adaptation of the systematic strategy, while the overlap setup creates two hypo/hypernymies and a synonymy as well, covering most of the disagreement cases.

For each data-set, we test the argumentation 500 times with a threshold randomly selected from the data-set dependent range presented in the previous paragraph. Figures A.3 and show, for each error threshold in absciss, the average initial and final synchronic agreement ratios and the average ratio of exchanged examples in ordinate, along with their standard deviations.

We observe that in both cases, the final synchronic agreement follows a same pattern: initially starting low with a great standard error, it stabilizes above 0.8 once the threshold reaches 5. However, the initial synchronic agreement has a significantly different profile for the two data-sets.

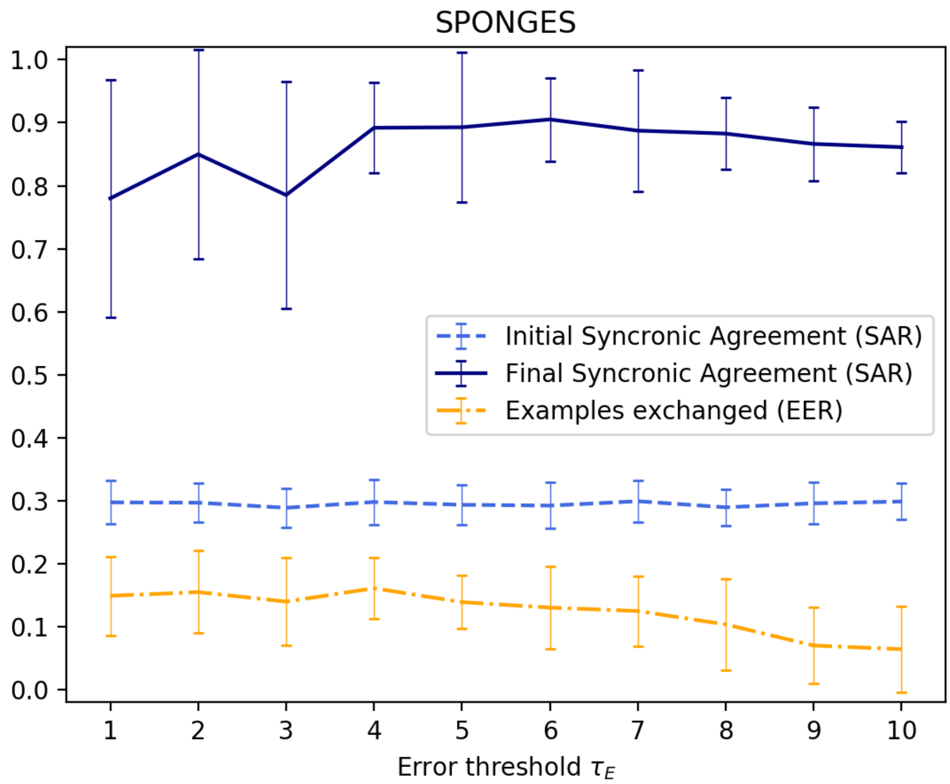


Figure A.3: Impact of the error threshold on synchronic agreement and examples exchanged, over an argumentation on overlaps in the Sponges data-set.

With the sponges data-set, we observe an initial SAR that stays at stable to 0.3 regardless of the threshold with the Sponges data-set – which is expected in a scenario of three concepts of the same size set up in an overlap, as it corresponds to one of the three initial concepts being the intersection of the two overlapping synonyms. On the contrary, the initial SAR in the Soybeans data-set displays multiple increasing and decreasing phases while the threshold varies, as the the threshold allows more or less classes of the data-set in the setup. For instance while we observe an initial SAR at 0.3 for a threshold ranging from 6 to 10, which corresponds to six usable concepts and therefore all of them involved in an overlap, the initial SAR rises at 0.4 above a threshold of 10 as only four concepts remains available for the overlap setup, three of them being used and the last one not causing significant disagreements to impact the initial SAR.

Finally, the number of examples exchanged is also impacted by the choice of the data-set. While the number of exchanged concepts decreases with the error threshold increasing in the Sponges data-set, it increases in the Soybean data-set – although not by a lot, and mostly when the threshold increases from one to five. A good explanation for this are the poor performances of the argumentation for thresholds ranging on the same values: on that range, the final SAR is significantly below what it is for the same thresholds in the case of Sponges. The inability to generalize, thus not creating good intensional definitions for small concepts and stopping the argumentation early limits the exchange of examples.

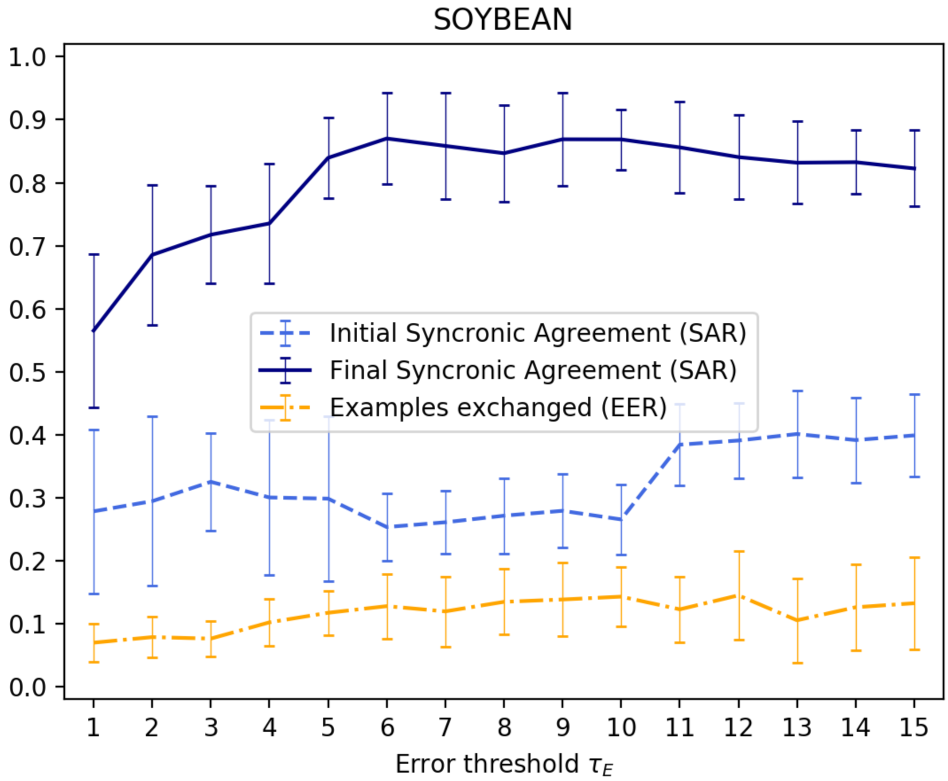


Figure A.4: Impact of the error threshold on synchronic agreement and examples exchanged, over an argumentation on overlaps in the Soybean data-set.

Appendix B

Messages

This appendix lists the different types of content (Table B.1) and performatives (Table B.2) that can be put in a message.

Class	Types	Notations
Elementary Values	Boolean, Integer, Double	b, i, d
Semiotic Components	Examples, Generalizations	e, g
Semiotic Elements	Extensional Definition, Sign, Intensional Definition	s, E, I, $s(C)$, $E(C)$, $I(C)$
Arguments	Root-Arguments, G-Arguments, E-Arguments	α
Identifiers	Concept Identifiers, Argument Identifiers	i, $i(C)$, $i(\alpha)$
Evaluations	R-Triplets, Pairing Relations	$r(C_i, C_j, U)$, (i_{-1}, i_0, i_1), $C_i r_U C_j$, $\equiv$, $\odot$, $\oslash$, $\otimes$

Table B.1: Different types of content available for messages. The different types are regrouped by classes in the left column. The right column gives some examples of content in their canonical notation.

Performative	Content	Description
Insert-Argument, Delete-Argument	α	Tells A_{-k} that the argument α has been inserted in/deleted from the argumentation.
Assert	$s(C)$, $id(C, A_k)$, $I(C)$	Informs the agent A_{-k} that A_k has a concept C, and shares its sign and intensional definition with A_{-k}.
Baptize	$s, id(C_i)$	Attributes a new sign s and a new $id(C_i)$ to the concept that is currently being created.
Debate	$id(C_i)$, $id(C_j)$	Proposes to resolve the disagreement caused by the relation between C_i and C_j.
Examples	E	Contains a set of examples E for A_{-k} to expends its local context.
Evaluation	$id(C_i)$, $id(C_j)$, $r(C_i, C_j, U)$	Shares the r-triplet of C_i and C_j with A_{-k}.
Intransitive	$id(C_1), \ldots, id(C_n)$	Informs the agent A_{-k} that from the point of view of A_k, $C_1 \ldots C_n$ are breaking the transitivity rule of the equivalence pairing relation.
Name	s, e	Tells A_{-k} that A_k associates the example e with the sign s.
Relation	$id(C_i)$, $id(C_j)$, r	Shares the pairing relation between C_i and C_j with A_{-k}.
Remove	$id(C_1)$, …, $id(C_n)$	Tells A_{-k} that all instances of the concepts $i(C_1)$, …, $i(C_n)$ should be removed from the argumentation.
Replace	s, $id(C)$	Tells A_{-k} that the sign of concept C is now s.
Root-Argument, Counter-Argument	α	Sends an argument α to A_{-k}.
Seize		Notifies A_{-k} that A_k will take care of an asymmetrical aspect of the argumentation during the next turn(s).
Self-Check		Asks A_{-k} to look for the pairing relations $R(S_{K,-k}, S_{H,-k}, U_O)$.
Size	i	Contains the size of a set of examples. Used to determine if a new concept has potentially enough examples to deserved to be created.
Vote	s, $id(C)$, d	Gives a support of value d to the fact that the concept C should have s for sign.

Table B.2: The different performatives available for messages. Each performative (left column) is presented with the type of content that is expected to be found in its instances (middle column). The role of each performative is presented in the right column.

Bibliography

Aıt-Kaci, H. (2007). Description logic vs. order-sorted feature logic. In *20th International Workshop on Description Logics DL'07*, volume 6, page 19.

Ait-Kaci, H. and Podelski, A. (1993). Towards a meaning of life. *The Journal of Logic Programming*, 16(3-4):195–234.

Amgoud, L. and Serrurier, M. (2007). Arguing and explaining classifications. In *International Workshop on Argumentation in Multi-Agent Systems*, pages 164–177. Springer.

Aras, R., Dutech, A., and Charpillet, F. (2004). Stigmergy in multiagent reinforcement learning. In *Fourth International Conference on Hybrid Intelligent Systems*, pages 468–469. IEEE.

Arcos, J. L. (1997). The noos representation language. *PhD thesis*.

Bourgne, G., El Fallah Segrouchni, A., and Soldano, H. (2007). Smile: Sound multi-agent incremental learning. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, page 38. ACM.

Bourgne, G., Soldano, H., and El Fallah-Seghrouchni, A. (2010). Learning better together. In *European Conference on Artificial Intelligence*, volume 215, pages 85–90.

Bundy, A. and McNeill, F. (2006). Representation as a fluent: An ai challenge for the next half century. *IEEE Intelligent Systems*, 21(3):85–87.

Carpenter, B. (2005). *The logic of typed feature structures: with applications to unification grammars, logic programs and constraint resolution*, volume 32. Cambridge University Press.

Chocron, P. and Schorlemmer, M. (2017). Vocabulary alignment in openly specified interactions. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1064–1072. International Foundation for Autonomous Agents and Multiagent Systems.

d'Aquin, M. (2009). Formally measuring agreement and disagreement in ontologies. In *Proceedings of the fifth International Conference on Knowledge Capture*, pages 145–152. ACM.

Davies, W. and Edwards, P. (1996). The communication of inductive inferences. In *Distributed artificial intelligence meets machine learning in multi-agent environments*, pages 223–241. Springer.

Doeben-Henisch, G. (2009). Humanlike computational learning theory. a computational semiotics perspective. In *AFRICON 2009*, pages 1–6. IEEE.

Euzenat, J. (2017). Interaction-based ontology alignment repair with expansion and relaxation. In *26th International Joint Conferences on Artificial Intelligence*, pages 185–191. AAAI Press.

Frake, C. (1962). The ethnographic study of cognitive systems. In Fishman, J. A., editor, *Readings in the Sociology of Language*, chapter 10, pages 434–446. De Gruyter.

Goldstone, R. L. and Rogosky, B. J. (2002). Using relations within conceptual systems to translate across conceptual systems. *Cognition*, 84(3):295–320.

Gómez, S. A. and Chesnevar, C. I. (2003). Integrating defeasible argumentation and machine learning techniques: A preliminary report. In *In Procs. V Workshop of Researchers in Comp. Science*. Citeseer.

Gudwin, R. and Gomide, F. (1997). An approach to computational semiotics. *National Institute of Standards & Technology Special Publication*, pages 467–470.

Gudwin, R. R. (1999). From semiotics to computational semiotics. In *Proceedings of the 9th International Congress of the German Society for Semiotic Studies, 7th International Congress of the International Association for Semiotic Studies*.

Guerrero, J. A., Gomes, A. S., and Gudwin, R. R. (1999). A computational tool to model intelligent systems. *Simpósio Brasileiro de Automação Inteligente*, pages 8–10.

Halpin, H. and McNeill, F. (2013). Discovering meaning on the go in large heterogenous data. *Artificial Intelligence Review*, 40(2):107–126.

Isaac, A., Trojahn, C., Wang, S., and Quaresma, P. (2008). Using quantitative aspects of alignment generation for argumentation on mappings. In *The 7th International Semantic Web Conference*, page 49.

Laera, L., Blacoe, I., Tamma, V., Payne, T., Euzenat, J., and Bench-Capon, T. (2007). Argumentation over ontology correspondences in mas. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, page 228. ACM.

Larson, J. and Michalski, R. S. (1977). Inductive inference of vl decision rules. *ACM SIGART Bulletin*, 63:38–44.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings*, pages 157–163. Elsevier.

Manzano, S., Ontanón, S., and Plaza, E. (2012). A case-based approach to mutual adaptation of taxonomic ontologies. In *International Conference on Case-Based Reasoning*, pages 226–240. Springer.

McNeill, F. and Bundy, A. (2007). Dynamic, automatic, first-order ontology repair by diagnosis of failed plan execution. *International Journal on Semantic Web and Information Systems*, 3:1–35.

Mitchell, T. M. (1982). Generalization as search. *Artificial intelligence*, 18(2):203–226.

Modi, P. J. and Shen, W.-M. (2001). Collaborative multiagent learning for classification tasks. In *Proceedings of the fifth International Conference on Autonomous Agents*, pages 37–38. ACM.

Možina, M., Žabkar, J., and Bratko, I. (2007). Argument based machine learning. *Artificial Intelligence*, 171(10-15):922–937.

Ogden, C. K. and Richards, I. A. (1923). *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*, volume 29. K. Paul, Trench, Trubner & Company, Limited.

Ontañón, S. and Plaza, E. (2007a). Case-based learning from proactive communication. In *16th International Joint Conferences on Artificial Intelligence*, volume 7, pages 999–1004.

Ontañón, S. and Plaza, E. (2007b). Learning and joint deliberation through argumentation in multiagent systems. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, page 159. ACM.

Ontañón, S. and Plaza, E. (2012). Similarity measures over refinement graphs. *Machine learning*, 87(1):57–92.

Ontañón, S. and Plaza, E. (2015). Coordinated inductive learning using argumentation-based communication. *Autonomous Agents and Multi-Agent Systems*, 29(2):266–304.

Prasad, M. N., Lesser, V. R., and Lander, S. E. (1996). Retrieval and reasoning in distributed case bases1. *Journal of Visual Communication and Image Representation*, 7(1):74–87.

Provost, F. J. and Hennessy, D. N. (1996). Scaling up: Distributed machine learning with cooperation. In *AAAI/IAAI, Vol. 1*, volume 1, pages 74–79. Citeseer.

Rahwan, I. and Simari, G. R. (2009). *Argumentation in artificial intelligence*, volume 47. Springer.

Rieger, B. B. (1997). Computational semiotics and fuzzy linguistics. In *Proceedings of the 1997 International Conference on Intelligent Systems and Semiotics*. Citeseer.

Sian, S. S. (1991). Extending learning to multiple agents: issues and a model for multi-agent machine learning (ma-ml). In *European Working Session on Learning*, pages 440–456. Springer.

Stone, P. and Veloso, M. (1998). Towards collaborative and adversarial learning: A case study in robotic soccer. *International Journal of Human-Computer Studies*, 48(1):83–104.

Stone, P. and Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383.

Trojahn, C., Moraes, M., Quaresma, P., and Vieira, R. (2008a). A cooperative approach for composite ontology mapping. In *Journal on data semantics*, pages 237–263. Springer.

Trojahn, C., Quaresma, P., and Vieira, R. (2008b). An argumentation framework based on confidence degrees to combine ontology mapping approaches. *International Journal of Metadata, Semantics and Ontologies*, 3(2):142–150.

Trojahn, C., Quaresma, P., and Vieira, R. (2012). Exploiting majority acceptable arguments for ontology matching. *International Journal of Artificial Intelligence*, 8:1–19.

Wardeh, M., Bench-Capon, T., and Coenen, F. (2009). Padua: a protocol for argumentation dialogue using association rules. *Artificial Intelligence and Law*, 17(3):183–215.

Wittgenstein, L. (2009). *Philosophical Investigations.* John Wiley & Sons.