Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

An Agent-Based Model of the Emergence and Transmission of a Language System for the Expression of Logical Combinations

Josefina Sierra-Santibáñez
Technical University of Catalonia, Campus Nord, Ed. Omega
c/ Jordi Girona Salgado 1-3, Barcelona 08034 Spain
E-mail: Maria.Josefina.Sierra@upc.edu

Abstract

This paper presents an agent-based model of the emergence and transmission of a language system for the expression of logical combinations of propositions. The model assumes the agents have some cognitive capacities for invention, adoption, repair, induction and adaptation, a common vocabulary for basic categories, and the ability to construct complex concepts using recursive combinations of basic categories and logical categories. It also supposes the agents initially do not have a vocabulary for logical categories (i.e. logical connectives), nor grammatical constructions for expressing logical combinations of basic categories through language. The results of the experiments we have performed show that a language system for the expression of logical combinations emerges as a result of a process of self-organisation of the agents' linguistic interactions. Such a language system is concise, because it only uses words and grammatical constructions for three logical categories (i.e. and, or, not). It is also expressive, since it allows the communication of logical combinations of categories of the same complexity as propositional logic formulas, using linguistic devices such as syntactic categories, word order and auxiliary words. Furthermore, it is easy to learn and reliably transmitted across generations, according to the results of our experiments.

1 Introduction

The question of the *origins and evolution of language* has received much interest in the last two decades. It has been approached from different disciplines such as anthropology, historical linguistics, evolutionary biology or artificial intelligence. In particular in artificial intelligence *agent-based models*, implemented and tested in computer simulations, have been used to study the emergence and evolution of language (Hurford, Studdert-Kennedy, and Kight 1998; Briscoe 2002; Lyon, Nehaniv, and Cangelosi 2007; Cartmill et al. 2014). Depending on whether these models emphasise the role of biological evolution or the role of cultural evolution, agent-based models are classified into two areas: *Biolinguistics*, which assumes that the structure of language is determined to a large extent by biological factors (Di Sciullo and Boeckx 2011); and *Evolutionary Linguistics*, which supposes language is primarily shaped by cultural forces (Minett and Wang 2005; Steels 2012).

The agent-based model proposed in this paper follows the evolutionary linguistics approach: It involves a population of autonomous software agents that interact with each other playing language games. A language game (Wittgenstein 1953) is typically an interaction between two agents, a speaker and a hearer. The speaker has a communicative goal, conceptualises the world for language, transforms this conceptualisation into an utterance, and communicates that utterance to the hearer. The hearer tries to parse the utterance, reconstruct its meaning and map it onto its own internal representation of the world. Speaker and hearer normally use extralinguistic means to determine the outcome of a language game and, depending on that outcome, employ different strategies to expand and adapt their internal languages in order to be more successful in future language games.

The agents in these models are initially endowed with a set of cognitive abilities that are assumed to be necessary for seeing the emergence of possible language systems that allow them to be successful in a language game (e.g. the ability to construct complex concepts, or to use and detect linguistic devices such as word order, syntactic categories or case markers). Then, they are made to play a series of language games, where they configure possible language systems and try them out. The goal of the experiments is to find out whether the population as a whole succeeds in the language game, i.e. communicates effectively, and to observe the conceptualisations and linguistic constructions that emerge in the population as a result of the processes of collective invention and negotiation, as well as the evolution over time of various macroscopic features of these language systems, such as the average size of the agents' grammars, or the similarity of their grammatical constructions.

Theories of language evolution study language change at two levels: that of language systems and that of language strategies. *Language systems* capture the regularity observed in some part of the vocabulary or the grammar of a language, for example, a system of colour terms, cases, tense-aspect distinctions or logical combinations. Language systems group a set of paradigmatic choices both on the side of meaning (the conceptual system) and on the side of form (the linguistic system). The *conceptual system* contains the semantic distinctions that are expressible in the language system and can therefore be used as building blocks for conceptualisation. The *linguistic system* includes the syntactic categories and grammatical constructions necessary to turn a conceptualisation into a concrete utterance (Steels 2011). Linguists call the approach underlying a language system a *language strategy*. A given language comprises many different language systems, which are closely integrated.

Agent-based experiments in evolutionary linguistics aim to explain how particular language systems and strategies may emerge and evolve. Examples of language systems that have been studied using agent-based models are: (1) *case systems* to express the role of participants in events (Steels 1998; Batali 1998); (2) *tense-aspect distinctions* (Gerasymova, Spranger, and Beuls 2012); (3) *agreement markers* to group words together (Beuls and Steels 2013); (4) *vocabularies* in co-evolution with *semantic categories* (Steels and Belpaeme 2005; Lara and Alfonseca 2000; 2002).

The rest of this paper is organised as follows. Firstly, we explain what we mean by a language system for the expression of logical combinations and specify the set of meanings that constitute its conceptual system. Secondly, we introduce the formalism the agents use to represent their grammars and the linguistic devices they may use to express the relation between each connective and its arguments in a sentence. Then, we describe the particular type of linguistic interaction that allows the agents to construct a common lexicon and a grammar, paying special attention to cognitive abilities for invention, adoption, repair, induction and adaptation. Next, we present the results of some experiments which study the emergence of a language system for the expression of logical combinations and its transmission across generations. Finally, we summarise the main contributions of the paper.

2 Conceptual System

We consider a scenario in which a group of agents try to communicate about *subsets of objects* of the set of all the objects in their particular context. We assume the agents have already developed a set of basic categories and a set of logical categories which allow them to build internal representations of the subset of objects that constitutes the topic of the language game they participate in at a given moment. In the agents' memories, basic categories are represented by propositional logic symbols such as `up` or `le`, which denote the propositions 'I am referring to the objects which are up' and 'I am referring to the objects which are to the left' respectively. Logical categories allow the agents to construct logical combinations of basic categories, which they represent internally by propositional logic formulas.

We also suppose that, at the beginning of a simulation run, the agents have a common vocabulary for referring to basic categories, but that they cannot express logical categories nor the logical formulas they can construct with them in their common language. This means that the agents are able to construct complex meanings such as 'I am referring to the objects which are either up or to the left, but not both', but they do not know how to communicate them. They should learn to express such meanings during the simulations.

The particular set of logical categories the agents can use to construct logical combinations of basic categories is the set of propositional logic connectives $\{\neg, \wedge, \vee\}$. The rest of the standard connectives of propositional logic (i.e. $\rightarrow$ and $\leftrightarrow$) can be expressed using formulas which recursively combine $\wedge, \vee$ and $\neg$ as follows: $A \rightarrow B \equiv \neg A \vee B$ and $A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$. In fact, every propositional logic formula is logically equivalent to another formula which uses only the connectives $\wedge, \vee$ and $\neg$.

The assumption that the agents initially have a set of basic categories, three logical categories, and a common vocabulary for basic categories is based on results from studies on the emergence of vocabularies for spatial concepts, colours or logical categories, in co-evolution with the semantic categories associated with these concepts, (Steels 1995; Steels and Belpaeme 2005; Sierra-Santibáñez 2002).

Previous works have also used experiments in which the topic of the language game is a subset of objects. However, the agents in (Beuls and Steels 2013) conceptualise the topic using a set of distinctive properties in which each property refers to a single object of the topic, and the role of agreement markers is to indicate which properties of the distinctive set refer to the same object. Consequently, the agents in (Beuls and Steels 2013) can only construct internal representations which are conjunctions of basic properties. The agents in (Sierra-Santibáñez 2014) conceptualise the topic using a propositional logic formula which combines basic categories and a Boolean function. This allows them to build a wider range of internal representations: negations, disjunctions, implications or any other type of logical formula that can be constructed using a single Boolean function of one or two arguments. But they are not able to construct logical formulas which contain more than one Boolean function. That is, their conceptual system does not include any recursive logical formulas. The conceptual system used in the present paper, in contrast, contains every propositional logic formula. Because in the present paper we precisely address the problem of the linguistic expression of recursive logical formulas combining basic categories and $\neg, \vee, \wedge$.

3 Linguistic System

Prolog Grammar Rules (Colmerauer et al. 1973; Pereira and Warren 1980) are used to represent the grammars constructed by the individual agents. The head of such rules is an atomic formula whose predicate symbol denotes a syntactic category (e.g. s for sentence) and whose arguments specify a number of aspects of the phrase described by that rule. In this paper the first argument conveys semantic information and the second one a score in the interval [0,1] that estimates the usefulness of the rule in previous communication. Semantic information can be a proposition, a logical connective of the form `or`, `and` or `not`, or a propositional logic formula constructed from the others. Logical formulas are represented using Lisp-like (McCarthy 1960) notation.

The following grammar uses *syntactic categories* and *word order* to express the relation between a connective and its arguments in a sentence. The number appearing in first place on the right hand side of grammar rules 3 and 6 indicates the position (i.e. first 1, second 2 or third 3) of the word associated with the main connective of the formula in the sentence with respect to the positions of the expressions associated with its arguments. This convention is necessary, because left recursive grammar rules cannot be used in Prolog. So the actual sentence generated by this grammar to express `[and,le,up]` is '2yizqarr', which can be parsed into 'izqyarr', where the word 'y' associated with the connective (`and`) is in the second position of the sentence.

$$s(\text{up}, S) \rightarrow \text{arr}, \{S \; is \; 0.7\} \quad (1)$$

$$s(\text{le}, S) \rightarrow \text{izq}, \{S \; is \; 0.25\} \quad (2)$$

$$s([P,Q,R], S) \rightarrow 2,\ c2(P,S_1), s(Q,S_2), s(R,S_3), \{S \; is \; S_1 \cdot S_2 \cdot S_3 \cdot 0.1\} \quad (3)$$

$$c2(\text{and}, S) \rightarrow \text{y}, \{S \; is \; 0.5\} \quad (4)$$

$$c2(\text{or}, S) \rightarrow \text{o}, \{S \; is \; 0.5\} \quad (5)$$

$$s([P,Q],S) \rightarrow 1, c1(P,S_1), s(Q,S_2)\{S \; is \; S_1 \cdot S_2 \cdot 0.6\} \quad (6)$$

$$c1(\text{not}, S) \rightarrow \text{no}, \{S \; is \; 0.5\} \quad (7)$$

This grammar, however, does not allow the agents in the present paper to unambiguosly communicate certain meanings. For example, it generates the same sentence, *'noarryizq',* to express formulas `[not,[and,up,le]]` and `[and,[not,up],le]`. Natural language, on the other hand, provides means to mark the difference between both meanings: `[not,[and,up,le]]` can be expressed as the English sentence *I'm referring to the objects which are not both up and to the left*, and `[and,[not,up],le]` as the sentence *I'm referring to the objects which are not up and which are to the left*. English does so by introducing an auxiliary word (i.e. *'both'*) to distinguish between both formulas, instead of new connectives (different from `and,or,not`) with their associated words and constructions. The auxiliary word *'both'* indicates that the conjunction following it forms an indivisible group and, therefore, that `not` (i.e. the negation) applies to the entire conjunction rather than just to the first element of such a conjunction. Thus a refined version of the grammar above would use sentence *'nogrouparryizq'* to express formula `[not,[and,up,le]]` and sentence *'noarryizq'* to mean `[and,[not,up],le]`, where *'group'* is an auxiliary word, invented by an agent, which plays the same role as *'both'* or *'either'* in English.

A similar problem comes up if we try to use the grammar above (extended by rule $s(\text{do}, S) \rightarrow \text{aba}, \{S \; is \; 0.1\}$) to express formula `[or,up,[and,do,le]]`, which can be paraphrased in English as *'the objects which are up, or down and to the left'*. It generates sentence *'arroabayizq'*, which is ambiguous with respect to the scope of words *'y'* and *'o'* in that sentence. For example, *'y'* could join expressions *'arroaba'* and *'izq'*, or expressions *'aba'* and *'izq'*. In the English sentence, a 'comma' is used to delimit the two main parts of the sentence. In speech the disambiguating role of punctuation marks is played by pauses, and in this work we will treat pauses as auxiliary words which indicate that the sentence following them forms an indivisible group and that the connectives in that group are within the scope of another connective. Thus a refined version of the grammar above would express `[or,up,[and,do,le]]` using sentence *'arropauseabayizq'*, where *'pause'* is also an auxiliary word invented or adopted by an agent.

The linguistic system constructed by the agents in the experiments described in this paper therefore uses three types of linguistic devices to express the relation between each connective and its arguments in a sentence: *syntactic categories, word order* and *auxiliary words.* Whereas the linguistic systems in (Steels 1998; Kirby 2002; Garcia and Steels 2014; Sierra-Santibáñez 2014) use only *syntactic categories* and *word order;* and those in (Gerasymova, Spranger, and Beuls 2012; Beuls and Steels 2013) use only tense-aspect or agreement markers[1]. The added difficulty of parsing and producing sentences with auxiliary words is, however, offset by the higher expressiveness of the present linguistic system, which allows the unambiguous communication of every propositional logic formula (see appendix A), and by the simplicity of its set of logical categories, which only contains logical connectives `and`, `or` and `not`.

4 Language Game

The main steps of the language game used in this paper, which is played by two agents randomly chosen from the population, are as follows:

1. The speaker chooses a logical formula (i.e. a meaning) from its set of conceptualisations of the subset of objects that constitutes the topic of the language game, generates or invents a sentence that expresses this formula, and communicates that sentence to the hearer.
2. The hearer tries to interpret the sentence communicated by the speaker. If it can parse it using its own grammar, it extracts a meaning (i.e. a logical formula); otherwise, the speaker communicates the formula it had in mind to the hearer, and the hearer adopts an association between that formula and the sentence used by the speaker.
3. Depending on the outcome of the language game speaker and hearer expand or adapt their grammars to be more successful in future language games.

A language game *succeeds* if the hearer can parse the sentence communicated by the speaker and if its interpretation of that sentence is logically equivalent to the formula the speaker had in mind; otherwise, the language game *fails*.

4.1 Generation, Invention and Repair

At the early stages of a simulation run the agents cannot use their own grammars to generate sentences for most meanings, because they all begin with a common lexicon for basic categories, but no lexicon for logical categories nor grammar rules. In order to let language get off the ground, the agents are allowed to invent new sentences for those meanings they cannot express using their grammars. A new sentence E for a propositional logic formula F of the form `[not,A]` or `[`$\otimes$`,A,B]` is invented as follows[2]: a new word is invented for its main connective (`not` or $\otimes$), an expression is generated for each argument (`A` and `B`), and the two or three expressions generated (depending on the type of formula) are concatenated in random order.

---

[1] Represented by affixes (groups of letters attached to words).

[2] If F is atomic, invention is not necessary because there exists a word in the common lexicon that expresses F. New words are sequences of three to six letters randomly chosen from the alphabet.

Once an agent can generate a sentence for a particular meaning using its grammar, it does not keep inventing new sentences for that meaning. It selects the sentence with the highest score from the set of all the sentences it can generate for that meaning. The *score of a sentence* (or a *meaning*) generated by a grammar rule is computed using the arithmetic expression on the right hand side of that rule, e.g. the score of sentence *'izqyarr'* –generated by rule 3 to express `[and,le,up]`– is computed multiplying the score of that rule (0.1) by the scores of rules 1, 2 and 4 (0.7, 0.25 and 0.5 respectively), which generate the words *'arr'*, *'izq'* and *'y'* associated with the constituents of that formula. The *score of a grammar rule* is the last number in the arithmetic expression that appears on the right hand side of that rule.

However, if an agent realises that the sentence generated by a grammar rule is ambiguous (e.g. sentence *'noarryizq'*, generated by rule 6, can be parsed as `[not,[and,up,le]]` or as `[and,[not,up],le]`), it applies a *repair operator* to that rule, replacing it with a set of rules (e.g. rules 8 and 9) that use additional checks and an auxiliary word invented or adopted by the agent to distinguish between the possible meanings of that sentence. For example, rule 8 refines rule 6 by adding check $Q \backslash= [\_,\_,\_]$, which ensures rule 8 is not applied to negations of conjunctions or disjunctions; and rule 9 introduces an auxiliary word (e.g. *'group'*) which indicates that the scope of *not* is the conjunction or disjunction following that auxiliary word.

$$s([P,Q],S) \to 1, c1(P,S_1), s(Q,S_2), \{Q \backslash= [\_,\_,\_], S\ is\ S_1 \cdot S_2 \cdot 0.1\} \quad (8)$$

$$s([P,[Q,R,T]],S) \to 1, c1(P,S_1), \mathrm{group}, s([Q,R,T],S_2), \{S\ is\ S_1 \cdot S_2 \cdot 0.1\} \quad (9)$$

Repair operations are also applied to grammar rules which generate ambiguous sentences to express conjunctions or disjunctions. For example, a *repair operator* may replace grammar rule 3 with rules 10, 11, 12 and 13. Rule 10 adds checks $Q \backslash= [\_,\_,\_]$ and $R \backslash= [\_,\_,\_]$ to ensure that the grammatical construction represented by rule 3 is not applied to conjunctions or disjunctions of other conjunctions or disjunctions. Rules 11, 12 and 13 introduce an auxiliary word invented or adopted by the agent (e.g. *'pause',*) to indicate that the conjunction or disjunction following that word is within the scope of the main connective of the sentence.

$$s([P,Q,R],S) \to 2, c2(P,S_1), s(Q,S_2), s(R,S_3), \{Q \backslash= [\_,\_,\_], R \backslash= [\_,\_,\_], S\ is\ S_1 \cdot S_2 \cdot S_3 \cdot 0.1\} \quad (10)$$

$$s([P,[Q_1,Q_2,Q_3],R],S) \to 2, c2(P,S_1), \mathrm{pause}, s([Q_1,Q_2,Q_3],S_2), s(R,S_3), \{R \backslash= [\_,\_,\_], S\ is\ S_1 \cdot S_2 \cdot S_3 \cdot 0.1\} \quad (11)$$

$$s([P,Q,[R_1,R_2,R_3]],S) \to 2, c2(P,S_1), s(Q,S_2), \mathrm{pause}, s([R_1,R_2,R_3],S_3), \{Q \backslash= [\_,\_,\_], S\ is\ S_1 \cdot S_2 \cdot S_3 \cdot 0.1\} \quad (12)$$

$$s([P,[Q_1,Q_2,Q_3],[R_1,R_2,R_3]],S) \to 2, c2(P,S_1), \mathrm{pause}, s([Q_1,Q_2,Q_3],S_2), \mathrm{pause}, s([R_1,R_2,R_3],S_3), \{S\ is\ S_1 \cdot S_2 \cdot S_3 \cdot 0.1\} \quad (13)$$

4.2 Interpretation and Adoption

In a language game the hearer tries to interpret the sentence communicated by the speaker using its own grammar. However, at the early stages of a simulation run the agents cannot parse most of the sentences communicated by speakers, because they begin with a common lexicon for basic categories but no grammar rules, except those that generate the words associated with basic categories. When this happens, the speaker communicates the formula F it had in mind to the hearer, and the hearer adopts an association between F and the sentence E used by the speaker, adding a new rule of the form $s(F,S) \to E, \{S\ is\ 0.1\}$ to its grammar. The initial score of the rules invented or adopted by the agents is 0.1.

Although in real life speakers do not communicate the exact meaning they have in mind to hearers, they usually provide some form of contextual or gestural feedback. Actually, some capacity for *intention-reading* seems to be necessary for language acquisition (Tomasello 2003). Most studies on grammar emergence and evolution also use language games in which the meaning intended by the speaker is communicated to the hearer (Beuls and Steels 2013; Gerasymova, Spranger, and Beuls 2012; Kirby 2002; Batali 1998; Sierra-Santibáñez 2014). The difficulty of studying language evolution in its full complexity leads researchers to focus on particular aspects of it, assuming results from research works addressing issues such as vocabulary and concept formation (Steels 1995; Lara and Alfonseca 2000; Sierra-Santibáñez 2001; Steels and Belpaeme 2005), or speech evolution (de Boer 2001; Zuidema and de Boer 2010).

Once an agent can parse a sentence using its grammar, it selects the meaning with the highest score from the set of all the meanings it can obtain for that sentence. However, if the meaning chosen by the hearer is not logically equivalent to the meaning the speaker had in mind, the speaker communicates the meaning it had in mind to the hearer and the hearer adopts an association between that meaning and the sentence used by the speaker.

4.3 Induction

Invention and adoption allow the agents to construct and learn associations between sentences and meanings. From these associations they induce grammatical constructions and lexical entries they incorporate to their grammars and use in subsequent language games to generate and interpret other sentences. Induction is performed applying three operations, which adapt the rules *simplification* and *chunk* in (Kirby 2002) to the type of grammars built by our agents. Let us see how these operations work with some examples. Suppose an agent adds rule 14 to its grammar.

$$\mathrm{s([and,le,up],S)} \to izqyarr, \{S\ is\ 0.7\} \quad (14)$$

Simplification of rules 1 and 14 allows replacing rule 14 with 15, which is more general because it can express meanings of the form `[and,le,R]`, where R is a variable.

$$\mathrm{s([and,le,R],S)} \to izqy, s(R,S_R), \{S\ is\ S_R \cdot 0.1\} \quad (15)$$

A second application of simplification to rules 15 and 2 allows replacing 15 with 16, where R and Q represent any meaning that can be expressed by a sentence, i.e. by an expression of syntactic category s.

$$\text{s([and,Q,R],S)} \rightarrow 2, y, s(Q, S_Q), s(R, S_R), \{S \ is \ S_Q \cdot S_R \cdot 0.1\} \tag{16}$$

If the agent's grammar contained rule 17, *chunk I* would allow the agent to generalise rules 16 and 17 as follows.

$$\text{s([or,Q,R],S)} \rightarrow 2, o, s(Q, S_Q), s(R, S_R), \{S \ is \ S_Q \cdot S_R \cdot 0.6\} \tag{17}$$

Chunk I creates a new syntactic category $c2$, and replaces rules 16 and 17 with rules 18, 19 and 20. Rule 18 generalises 16 and 17, because it can express meanings of the form `[P,Q,R]`, where `P` is any logical category that can be expressed by a word of syntactic category $c2$. Rules 19,20 state that words *'y'* and *'o'* belong to syntactic category $c2$.

$$\text{s([P,Q,R],S)} \rightarrow 2, c2(P, S_1), s(Q, S_2), s(R, S_3), \{S \ is \ S_1 \cdot S_2 \cdot S_3 \cdot 0.1\} \tag{18}$$

$$\text{c2(and,S)} \rightarrow y, \{S \ is \ 0.1\} \tag{19}$$

$$\text{c2(or,S)} \rightarrow o, \{S \ is \ 0.1\} \tag{20}$$

Chunk II allows adding words to syntactic categories created by chunk I. Chunk II replaces a grammar rule for a specific connective (e.g. 21) with a new word of a syntactic category created by chunk I, if the rules for the connective and the syntactic category (e.g. 18) are sufficiently similar. Suppose again the agent added rule 21 to its grammar.

$$\text{s([if,Q,R],S)} \rightarrow 2, si, s(Q, S_Q), s(R, S_R), \{S \ is \ S_Q \cdot S_R \cdot 0.1\} \tag{21}$$

Chunk II, applied to 21 and 18, would replace 21 with 22.

$$\text{c2(if,S)} \rightarrow \text{si, \{S is 0.1\}} \tag{22}$$

Although both induction and repair operations manipulate the set of rules in an agent's grammar, there are important differences between them. The former are applied whenever a new association is added to an agent's grammar. They replace a single rule or a pair of them with a grammar rule that is more general, i.e. that can be used to express more meanings. The repair operations used in the agent-based model proposed in this paper, on the other hand, do not generalise existing rules, as induction operations do. Instead, they specialise grammar rules which are too general and can generate ambiguous sentences. Induction rules operationalise the *pattern-finding* ability children develop according to *the usage-based theory of language acquisition* (Tomasello 2003). Whereas repair operations implement cognitive processes associated with a particular type of grammar invention and overgeneralisation examples reported in studies of children language acquisition (Tomasello 2006).

4.4 Adaptation

Coordination of the agents' grammars is necessary, because different agents can invent different words to refer to the same connective and they may concatenate the expressions associated with the components of a formula in different orders when they try to express it as a sentence. Although (Steels 1998; Kirby 2002) also study the acquisition of word-order based grammar, they do not address the issue of coordination. Because the populations in these works consist only of two agents, which necessarily share the same history of linguistic interactions. In the experiments discussed in the present paper coordination is achieved through a process of self-organisation of the agents' linguistic interactions, which takes place when these agents adapt their preferences for vocabulary and grammatical constructions to those they observe are used more often by other agents.

The agents adapt the scores of their grammar rules (i.e. their preferences for vocabulary and grammatical constructions) at the last step of a language game, when the speaker communicates the meaning it had in mind to the hearer, and only in the case in which the speaker can generate at least one sentence for the meaning it is trying to communicate using its grammar and the hearer can parse the sentence communicated by the speaker. If an agent can generate several sentences for expressing a given meaning, it chooses the sentence with the highest score, and temporarily stores the other sentences in a set called *competing sentences*; similarly, if it can obtain several meanings for a sentence, it selects the meaning with the highest score and stores the other meanings in a set called *competing meanings*. In a language game only the agent playing the role of hearer adapts the scores of its grammar rules. However, as all the agents in the population play both the role of speaker and that of hearer in different language games, all of them have ample opportunity to adapt the scores of their grammar rules during a simulation.

If the meaning interpreted by the hearer is logically equivalent to the meaning the speaker had in mind, the hearer adjusts the scores of its grammar rules both at the level of interpretation and at that of generation: 1) It increases the scores of the rules it used for obtaining the meaning the speaker had in mind and decreases the scores of the rules it used for obtaining competing meanings. 2) It tries to simulate what it would have said if it had been in the speaker's place, i.e. it tries to express the meaning the speaker had in mind using its own grammar; and it increases the scores of the rules that generate the sentence used by the speaker and decreases the scores of the rules that generate competing sentences.

If the meaning interpreted by the hearer is not logically equivalent to the meaning the speaker had in mind, the hearer decreases the scores of the rules it used for obtaining its interpretation of the sentence used by the speaker, and adopts an association between the sentence and the meaning used by the speaker if it cannot obtain such an association using its own grammar.

5 Experiments

The agent-based model proposed in this paper has been implemented in Prolog (Bueno et al. 1997) and tested on a series of experiments which study both the emergence of a language system for the expression of logical combinations and its transmission across generations.

In the first experiment, which studies language emergence, the agents begin with a common lexicon for basic categories[3] but no grammar rules. Then they play 6060 language games about propositional logic formulas constructed

---

[3] (Gerasymova, Spranger, and Beuls 2012; Beuls and Steels 2013) also use simulations with software agents and initialise the agents with a common vocabulary for basic categories.

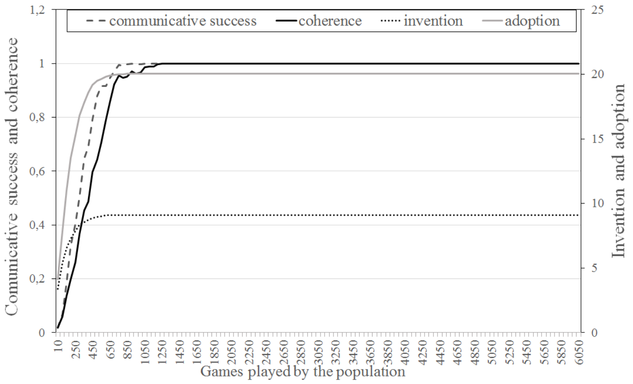

Figure 1: Evolution of communicative success, coherence, invention and adoption in an experiment with ten agents.

from the common set of basic categories. These formulas may have a non-trivial recursive structure, i.e. connectives `and`, `or` and `not` can be applied to non-atomic formulas.

Figure 1 shows the evolution over time of four measures that monitor the population's global performance. *Communicative success* is the average of successful language games in the last ten games played by the agents. The population reaches full communicative success (i.e. 1.0) in 1050 language games, i.e. after each agent has played 210 games on average. *Coherence* measures the similarity of the agents' grammars. It is the average of language games in which: (1) the hearer understands correctly the sentence communicated by the speaker, and (2) the hearer would use the same sentence as the sentence used by the speaker to communicate the meaning the speaker had in mind. Coherence increases slightly more slowly than communicative success. Full coherence (1.0) is reached in 1250 games (i.e. 250 games per agent). In a similar experiment described in (Sierra-Santibáñez 2014), in which the agents play language games only about non-recursive logical formulas (constructed from basic categories and a set of 11 unary and binary Boolean functions), full communicative success is reached later, in 1950 games, and full coherence much later, in 4600 games.

Invention (respectively *adoption*) is the number of sentences invented (respectively adopted) by an agent in past language games. The results shown in figure 1 are the average values of invention (adoption) for a population of ten agents in ten simulation runs[4]. Invention grows rapidly during the first 700 games, reaching a maximum average value of 9.11 inventions per agent. At that point the agents stop inventing new sentences. The average number of sentences adopted per agent keeps growing during the first 950 games, reaching a maximum average value of 20.05 adoptions per agent. In (Sierra-Santibáñez 2014) the agents invent less constructions (5.14 inventions per agent), and they clearly adopt more constructions (31.03 adoptions per agent) and for a longer period (during the first 1900 games)[5].

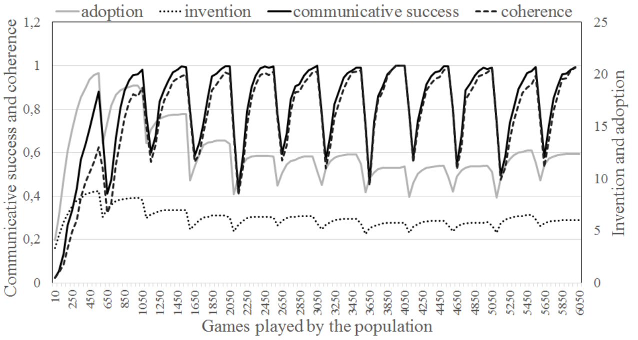

Figure 2: Evolution of communicative success, coherence, invention and adoption in an experiment studying language transmission: $\frac{1}{3}$ of the population renewed every 500 games.

Figure 2 shows the results of an experiment studying language transmission across generations. The agents in the population are divided into three groups: the elder, the adults and the young. Every 500 games the elder (about one third of the population) are replaced with new agents which have neither a vocabulary for logical connectives nor grammar rules. The previous adults become the elder, the young the adults and the new agents the younger generation. As a consequence the population is completely renewed every 1500 games. The four measures decrease whenever new agents are introduced in the population, but they catch up well before the next generation of agents is introduced. Communicative success typically reaches values over 0.99 and coherence over 0.98. The final average values of invention (6,04) and adoption (12.37) per agent are lower than in the experiment with a single generation (9.11 invention and 20.05 adoption). This is due to the fact that new agents learn a language already established in the population (i.e. the language transmitted from generation to generation in the experiment), which uses fewer variations for expressing a given meaning. Comparing the communicative success and coherence curves in figure 2 with the corresponding curves in a similar experiment described in (Sierra-Santibáñez 2014), we can observe that local maxima are less sharp and more rounded in the former than in the later. This suggests that high values for communicative success and coherence are reached earlier and for longer periods in the present experiment, and thus that language transmission is faster and more reliable. Furthermore, the score values of preferred constructions in figure 3 are clearly more stable and higher than the scores of preferred constructions in (Sierra-Santibáñez 2014), which are always lower than 0.6.

Figure 3 shows the evolution of the scores of preferred constructions (i.e. with average scores greater than 0.7) in a simulation run of the experiment studying language transmission. Each line in the graph displays the evolution of the average score of one particular construction for all the agents in the population. Constructions are labelled with the word used to express the main connective (`and,or,not`) of the formula expressed by that particular construction, preceded by a number which indicates the position of that word in the sentence. During the whole simulation, which involves 12 generations, the agents invent a total of 29 constructions for expressing conjunctions, 21 for disjunctions and 33 for

---

[4] (Vogt 2005; Gerasymova, Spranger, and Beuls 2012; Beuls and Steels 2013) use the same population size.

[5] The higher value of invention in the present experiment with respect to (Sierra-Santibáñez 2014) might be due to the fact that repair operations are counted as inventions in the present experiment.

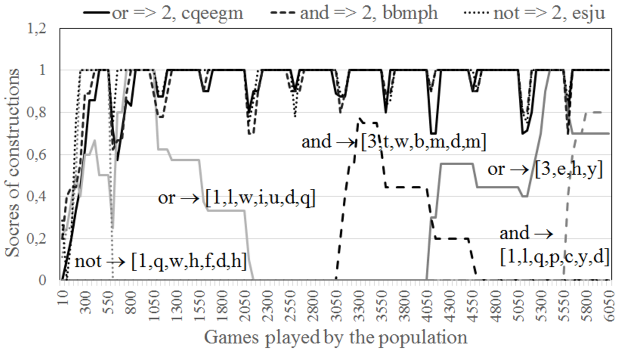

Figure 3: Score evolution of preferred constructions for expressing formulas whose main connective is `or` (solid lines), `and` (dashed) or `not` (dotted) in a simulation run of the experiment studying language transmission.

negations. These constructions use different words for expressing a particular connective, and place these words in different positions of the sentence (first, second or third). However, once constructions *not* $\rightarrow$ *2,esju*, *and* $\rightarrow$ *2,bbmph* and *or* $\rightarrow$ *2,cqeegm* begin to be preferred by the agents in the first generation (after 450 games), their average scores keep reaching the maximum value (1.0) in succeeding generations, although they may occasionally coexist with some synonyms (see constructions for `not` and `or`) during short periods. This means that the language system constructed by the first generation is transmitted without change to succeeding generations up to the last one in the experiment.

6 Conclusions

This paper proposes an agent-based model of the emergence and transmission of a language system for the expression of logical combinations of propositions. The model has been implemented in Prolog and tested conducting a series of experiments in which a group agents try to communicate about subsets of objects characterised by logical combinations of categories of the same complexity as propositional logic formulas. The results of the experiments we have performed show that a language system for the expression of logical combinations emerges as a result of a process of self-organisation of the agents' linguistic interactions.

Such a language system is concise, because it only uses words and grammatical constructions for three logical categories (`and,or,not`). From a logical combination point of view, it is also more expressive than the language systems constructed in (Kirby 2002; Beuls and Steels 2013; Sierra-Santibáñez 2014). Since it allows the communication of concepts of the same complexity as propositional logic formulas, using linguistic devices such as syntactic categories, word order and auxiliary words to specify the relation between each connective and its arguments in a sentence. Furthermore, it is easy to learn (having only one unary and two binary commutative connectives); and reliably transmitted across generations in the simulations we have performed.

This work also contributes a set of Prolog software tools that allow conducting experiments on language evolution, monitoring the population's performance, and dynamically analysing the individual grammars constructed by the agents.

Acknowledgments

The author would like to thank Manuel Alfonseca, Diane Fleming, Luc Steels and Glyn Morrill for their help at different stages during the realisation of this work. The research leading to these results received funding from the Spanish Ministry of Economy and Competitiveness under project TIN2011-27479-C04-03 and the Catalan Agency for Research and University Funding under project SGR-890.

A Expressiveness of the Linguistic System

It is clear that the invention, adoption, repair and induction operations proposed in this paper allow the agents to construct a prefix grammar (e.g. the grammar consisting of rule $s([P,Q,R],S) \rightarrow 1, c2(P,S_1), s(Q,S_2), s(R,S_3), \{S\ is\ S_1 \cdot S_2 \cdot S_3 \cdot 0.1\}$ and rules 1, 2, 4, 5, 6 and 7 in section 3), and that this prefix grammar –which does not use auxiliary words– can generate unambiguous sentences for every propositional logic formula.

Linguists have observed a tendency to minimise the distance between related words/phrases in natural language. This might be so, because the number of unresolved dependencies accumulated during parsing (Morrill 2010) is larger when related words/phrases are distant from each other. Many natural languages do not use prefix notation (where connective scope is unambiguous) to express conjunctions or disjunctions. A possible explanation is that in prefix notation the distance between a connective and its second argument is proportional to the number of connectives in the sentence. Instead, these languages place words expressing conjunction or disjunction between the expressions associated with their arguments, although this form of infix notation generates the ambiguity problems described earlier.

An example of a typical infix grammar the agents can construct in our experiments consists of grammar rules 1, 2, 4, 5 and 7 in section 3 and rules 8 to 13 in section 4.1. This infix grammar allows expressing every propositional logic formula unambiguously, because: 1) It can generate a sentence for every propositional logic formula; 2) Every sentence u generated by this grammar can be mapped onto an equivalent prefix sentence $t(u)$ with the same meaning which does not contain any auxiliary words as follows. Reading u from left to right we assign consecutive natural numbers i to each occurrence of an auxiliary word (*'group'* or *'pause'*) until we find an occurrence of a word expressing conjunction or disjunction (*'y'* or *'o'*). We replace then the occurrence of *'group'* or *'pause'* associated with the last natural number assigned i with the word expressing conjunction or disjunction found, eliminate the occurrence of the word expressing conjunction or disjunction found, set the last natural number assigned to $i-1$; and continue assigning numbers from $i-1$ to occurrences of 'group' and *'pause'*, and moving occurrences of *'y'* and *'o'* to their corresponding prefix positions until we reach the end of the sentence. There may be a single occurrence of a word expressing conjunction or disjunction that will not be associated with a previous occurrence of *'group'* or *'pause'*. If we find this word the value of i will be zero and we will move this word to the beginning of the sentence, because it represents the main connective of the formula. The sentence $t(u)$ generated in this manner is unique and unambiguous, because it is in prefix notation. Therefore, it guarantees that the sentences generated by the infix grammar are also unambiguous.

References

Batali, J. 1998. Computational simulations of the emergence of grammar. In *Approaches to the Evolution of Language: Social and Cognitive Bases*, 405–426. Cambridge Univ Press.

Beuls, K., and Steels, L. 2013. Agent-based models of strategies for the emergence and evolution of grammatical agreement. *PLoS ONE* 8(3):e58960.

Briscoe, T., ed. 2002. *Linguistic Evolution through Language Acquisition: Formal and Computational Models*. Cambridge: Cambridge University Press.

Bueno, F.; Cabeza, D.; Carro, M.; Hermenegildo, M.; López-García, P.; and Puebla, G. 1997. The Ciao Prolog system. reference manual. Technical Report CLIP3/97.1, School of Computer Science, Technical University of Madrid (UPM). Available from http://www.clip.dia.fi.upm.es/.

Cartmill, E.; Roberts, S.; Lyn, H.; and Cornish, H., eds. 2014. *Evolution of Language, Proceedings of the Tenth International Conference EVOLANG*. World Scientific.

Colmerauer, A.; Kanoui, H.; Pasero, R.; and Roussel, P. 1973. Un système de Communication Homme-machine en Francais, Research Report. Technical report, Groupe Intelligence Artificielle, Université Aix-Marseille II, France.

de Boer, B. 2001. *The Origins of Vowels Systems*. Oxford University Press.

Di Sciullo, M., and Boeckx, C., eds. 2011. *The Biolinguistic Enterprise. New perspectives on the Evolution and Nature of the Human Language Faculty*. Oxford University Press.

Garcia, E., and Steels, L. 2014. Strategies for the emergence of first-order phrase structure. In *Evolution of Language, Proceedings of the Tenth International Conference EVOLANG*, 50–57. World Scientific.

Gerasymova, K.; Spranger, M.; and Beuls, K. 2012. A language strategy for aspect: Encoding aktionsarten through morphology. In *Experiments in Cultural Language Evolution*, 257–276. Springer.

Hurford, J.; Studdert-Kennedy, M.; and Kight, C., eds. 1998. *Approaches to the Evolution of Language: Social and Cognitive Bases*. Edinburgh University Press.

Kirby, S. 2002. Learning, bottlenecks and the evolution of recursive syntax. In *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, 96–109. Cambridge University Press.

Lara, J., and Alfonseca, M. 2000. Some strategies for the simulation of vocabulary agreement in multi-agents communities. *Journal of Artificial Societies and Social Simulation* 3(4). http://jasss.soc.surrey.ac.uk/3/4/2.html.

Lara, J., and Alfonseca, M. 2002. The role of oblivion, memory size and spatial separation in dynamic language games. *Journal of Artificial Societies and Social Simulation* 5(2). http://jasss.soc.surrey.ac.uk/5/2/1.html.

Lyon, C.; Nehaniv, C.; and Cangelosi, A., eds. 2007. *Emergence of Language and Communication*. Lecture Notes in Computer Science. Berlin: Springer-Verlag.

McCarthy, J. 1960. Recursive functions of symbolic expressions and their computation by machine, part I. *Communications of the ACM* 3(4):184–195.

Minett, J., and Wang, W., eds. 2005. *Language Acquisition, Change and Emergence: Essays in Evolutionary Linguistics*. City University of Hong Kong Press.

Morrill, G., ed. 2010. *Categorial Grammar: Logical Syntax, Semantics and Processing*. Oxford University Press.

Pereira, F., and Warren, D. 1980. Definite clause grammars for language analysis a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence* 13(3):231–278.

Sierra-Santibáñez, J. 2001. Grounded models as a basis for intuitive reasoning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI-2001*, 401–406.

Sierra-Santibáñez, J. 2002. Grounded models as a basis for intuitive and deductive reasoning: The acquisition of logical categories. In *Proceedings of the European Conference on Artificial Intelligence, ECAI-2002*, 93–97.

Sierra-Santibáñez, J. 2014. An agent-based model studying the acquisition of a language system of logical constructions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI-2014*, 350–357. AAAI Press.

Steels, L., and Belpaeme, T. 2005. Coordinating perceptually grounded categories through language: a case study for colour. *Behavioral and Brain Sciences* 28:469–529.

Steels, L. 1995. A self-organizing spatial vocabulary. *Artificial Life* 2(3):319–332.

Steels, L. 1998. The origins of syntax in visually grounded robotic agents. *Artificial Intelligence* 103(1-2):133–156.

Steels, L. 2011. Modeling the cultural evolution of language. *Physics of Life Reviews* 8:339–356.

Steels, L., ed. 2012. *Experiments in Cultural Language Evolution*. John Benjamins.

Tomasello, M. 2003. *Constructing a Language: a Usage-Based Theory of Language Acquisition*. Harvard Univ Press.

Tomasello, M. 2006. Acquiring linguistic constructions. In *Handbook of child psychology*. Wiley Online Library.

Vogt, P. 2005. The emergence of compositional structures in perceptually grounded language games. *Artificial Intelligence* 167(1-2):206–242.

Wittgenstein, L. 1953. *Philosophical Investigations*. New York: Macmillan.

Zuidema, W., and de Boer, B. 2010. Multi-agent simulations of the evolution of combinatorial phonology. *Adaptive Behavior* 18(2):141–154.