Norm Emergence in Complex Ambiguous Situations

Kiran Lakkaraju
Department of Computer Science
The University of Illinois, Urbana-Champaign

Les Gasser
Graduate School of Library and Information Science
The University of Illinois, Urbana-Champaign

Abstract

Our general objective is to explain how norms can emerge in complex, ambiguous situations: settings with large and complex spaces of normative options over which populations may try to agree using only limited, indirect knowledge of each others' currently preferred options, possibly gained through limited interaction samples.

We study this process using the concrete example of agents developing and using common languages. Language can be viewed as an inherently distributed information and representation system. Because of this, it serves as a model problem for studying central issues in many kinds of distributed information systems, and norms are one such issue. Language is *inherently normative*. First, communicative language requires agreement—conventions—on many language aspects (such as ontological units, grammar, lexicon, morphology, etc.). Second, accurate communication is valuable. The value of successful communication translates to value for the conventionalization of language; this value in turn creates the decentralized normative force that drives agents to obey linguistic conventions. In this way, linguistic conventions become normative constraints on possible communication options, since obeying them increases communicability and its resulting communication payoffs.

None of this means that convergence to linguistic norms is easy; in fact it presents novel issues not yet well understood. We show how ambiguity arises in language convergence, and describe a variety of techniques to resolve that ambiguity. We focus on one particular technique, *text based learning*. We show that it significantly reduces the amount of effort required for linguistic norms to emerge, and we show how it is an instance of a general norm-convergence technique.

Introduction

Norms have been proposed as one type of decentralized organizing principle for multi-agent systems. Loosely speaking, norms are collective behavioral conventions that have an effect of constraining, structuring, and making predictable the behaviors of individual agents—that is, they are conventions that can exert a *normative force* that shapes individual agent behaviors—removing some behavioral possibilities from consideration and encouraging others—without a centralized "enforcement agency." Norms are important in MAS research for two primary reasons: 1) their decentralized behavior-shaping effect is a potentially scalable, adaptive control mechanism, and 2) human groups and societies have been analyzed using norm concepts, so computational normative models may provide concrete embodied theories of this aspect of social systems.

The norm concept exists in numerous disciplines, including economics, sociology, and multi-agent systems. (Opp 2001) reviews much of this literature, providing a useful summary and synthesis of definitions of norms. His synthesis comprises three components: 1) widely shared expectation of normative behavior, 2) probability of sanction (i.e. negative utility) if norm is violated, and 3) norm-driven collective behavioral regularity. (Note that there are other non-normative causes of behavioral regularity, such as correlated preferences.)

In this paper we explore *language* as a normative behavior, and treat the emergence and evolution of language as a problem of collective emergence and evolution of several classes of linguistic norms—specifically norms of lexical (word meaning) and structural (grammatical) semantics. Within this specific context of language, we investigate two key general questions about norms:

- How can norms effectively emerge through a distributed process of agent interaction in large, complex normative spaces with limited knowledge (ambiguity)?
- What are the mechanisms through which "normative force" is applied?

Complexity of Normative Spaces and Limited Knowledge

The language case introduces three intricacies that enrich the general study of norms:

Complexly Structured Option States A language has many interactions between its elements. For example, lexical (word-meaning) and structural (syntactic, grammatical) features of language interact: choosing a grammar will establish certain lexical constraints, and conversely determining a lexical role for a word will constrain the possible grammars that fit that interpretation. Thus language as a normative case is quite different from other convention phenomena studied earlier (e.g. (Shoham & Tennenholtz 1997)) that have a simple binary normative option space. Here we need to treat lexicon and grammar as complex interdependent norm components, and to find ways of making their joint emergence tractable for agents - our approach to this is one of the central insights of the paper.

Large Normative Option Space Certain aspects of a language's lexicon can change independently from its grammar - for example, words in an established grammatical category ("noun," "verb") can be added to or deleted from the language without changing its grammar. Thus the space of possible options for interpreting an utterance changes multiplicatively with changes to the lexical and/or grammatical complexity. For this reason, there is an enormous number of possible combined lexical/grammatical interpretations for a given utterance when the language is even moderately complex. A population must converge on a shared lexical and grammatical convention within this very large space—again, a significant difference from earlier studies with binary normative spaces.

Ambiguity and Imperfect Knowledge Agents have no direct access to others' preferred linguistic choices (normative states). Instead they must infer other agents states on the basis of limited linguistic samples (utterances) possibly coupled with a limited number of shared situations from which to "ground" semantic choices. In general, this limits agents to imperfect knowledge of collectively-preferred normative options, creating ambiguity when they try to converge. Again, this differs significantly from binary option spaces in which knowledge of other agents' states can be inferred perfectly: $\neg 1 = 0$ and $\neg 0 = 1$.

Normative Force in Language

In Opp's conceptualization of norms (Opp 2001) the issue of normative force is covered by reference to "probability of sanctions." A norm carries some probability of sanction (negative utility) when violated. Our model of how normative force appears in the context of language use is based on the notion of *communicability*, or mutual intelligibility between agents. We assume that the ability to communicate via language confers an advantage or payoff to agents. For example, linguistic competence may give agents access to otherwise-inaccessible knowledge, or it may allow agents to coordinate more effectively. Both of these may improve performance on (increase payoff from) collective tasks. The *communicability* of two agents, speaker and hearer, can be quantified as the fraction of all possible situations that the speaker can accurately communicate to the hearer (see below). Average population-wide communicability is increased by converging more closely to a common language, so that when two random agents interact over a random situation, their expected task payoff is higher. Since reliable communication has value, and we assume agents want to increase their payoffs from tasks, communicability provides the decentralized normative force encouraging the behavior of more closely conforming to widely-shared languages. Widely held linguistic conventions become normative constraints on possible communication options because

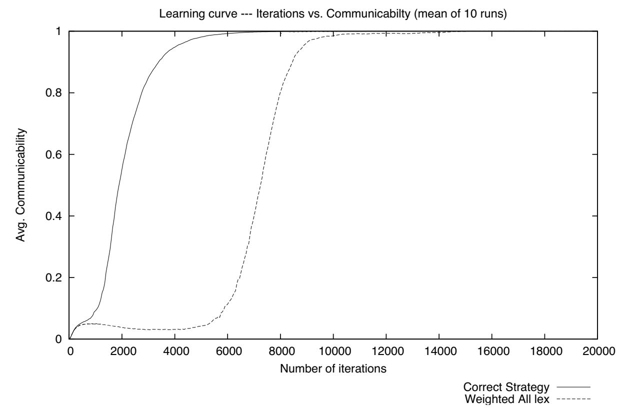

Figure 1: Convergence curve for 20 agents and 30 meanings. There are 27 object predicates and 3 event predicates.

they encourage communicability and thus higher task payoffs. Violation of linguistic norms leads to lower payoffs, which can be interpreted as sanctions. When agents act outside the linguistic norm, their lower communicability yields lower payoffs, i.e. sanctions.

A major aspect of norm research is focused on the *internalization* of norms, which naturally flows from our modelization of conventions and normative force—our work follows in the vein of (Shoham & Tennenholtz 1997; Savarimuthu *et al.* 2007). Reification or *immergence* of norms to provide second-order behavioral control is not treated here; for an account of it see, e.g., (Andrighetto *et al.* 2007).

The Ambiguity Problem

One of the aspects of the complex setting of language is *ambiguity* due to imperfect knowledge. Ambiguity arises in situations where agents utilize imperfect knowledge of another agents behaviors. When agents are faced with imperfect knowledge the ability for a norm to develop will be affected.

The effect of ambiguity is clearly present in the case of language convergence. In cases where a population of agents must form a linguistic norm imperfect knowledge causes a significant increase in the time to form a common language.

Figure 1 shows the time to convergence for a population of agents. The x axis indicates time (measured in number of interactions) and the y access indicates the average communicability of the population. When the population has reached an average communicability of 1 a linguistic norm has been achieved.

The left hand curve is the situation where agents are given perfect knowledge of another agents linguistic behaviors. A linguistic norm is achieved in around 6000 iterations. This is the optimal case, where we as the designers, have provided the agents with the correct information.

The curve on the right is the situation with imperfect knowledge. We can see that in the case of imperfect knowledge there is a significant increase in the amount of time to convergence. A norm is only achieved in 12,000 iterations. The increase is because of the errors agents make due to ambiguity as to the correct linguistic behavior of other agents.

Our objective is to identify and develop techniques to address the ambiguity problem and bridge the gap between the perfect case and the imperfect knowledge case.

The rest of the paper is organized as follows. In the "Settings" section we describe the setting in which we explore language convergence and ambiguity. The "Imperfect Knowledge Leads to Ambiguity" section illustrates how ambiguity arises in the language case, and general methods for resolving such ambiguity. The "Text-Based Ambiguity Reduction Strategies" section and the "Text Based Observation Games" section go over one technique, the "Text Based Observation Game" that we have developed. The "Results" section evaluates how well the text based observation game reduces ambiguity.

Setting

Agents

The simulation setting contains a population of agents, denoted $\mathcal{A}$.

The agents are taking part in a *Description Task*: their objective is to describe scenes from the environment to each other. Describing the relation between two or more objects is a crucial prerequisite to more complex interactions concerning coordination, cooperation and negotiation.

The agents can perceive and represent scenes from the environment (to be described below) internally. Agents also have a set of linguistic behaviors that they use to transform their internal representation of scenes into communicable forms.

Agents are endowed with a memory that keeps track of scenes they have seen. No memory capabilities beyond this are assumed.

The Environment

Our environment contains two types of entities:

Objects Objects are physical entities in the world, such as cats, dogs, trees, etc.

Actions Actions are activities/events that occur with respect to objects in the world. There can be no action without one or more objects, e.g. the action of chasing requires an object as a chaser and an object to be chased.

A *scene* drawn from the environment is a relationship between two objects and an action. For example, a scene would be a dog chasing a cat.

The environment is strategically simple in order to focus on the ambiguity problem. We find that even with such a simple environment substantial ambiguity will arise.

All agents can accurately perceive the environment and represent scenes in the environment using a First Order Logic (FOL) based representation. For instance, the scene "Dog chases cat" is encoded:

$$\begin{aligned} &\mathit{Dog}(\mathsf{x}) \wedge \mathit{Cat}(\mathsf{y}) \wedge \mathit{Chase}(\mathsf{z}) \wedge \mathit{Agent}(\mathsf{x}) \wedge \mathit{Patient}(\mathsf{y}) \\ &\qquad \wedge \mathit{Event}(\mathsf{z}) \wedge \mathit{TransitiveAction}(\mathsf{z}, \mathsf{x}, \mathsf{y}) \end{aligned}$$

This scene contains three entities: a dog ($\mathit{Dog}(\mathsf{x})$), a cat ($\mathit{Cat}(\mathsf{y})$) and the "chasing" event ($\mathit{Chase}(\mathsf{z})$).

Variables represent entities in the environment. Single arity predicates represent properties of the entity, i.e., $\mathit{Red}(\mathsf{x})$ means the entity x refers to has the property of being red. $\mathit{Chase}(\mathsf{y})$ means that the entity referred to by y has the property of being a chase event.

Multi-arity predicates refer to relations between entities in the world. We only utilize one relation: $\mathit{TransitiveAction}(\mathsf{x}, \mathsf{y}, \mathsf{z})$. This predicate indicates that entity y is performing the action referred to by x upon z.

Clearly the ordering of TransitiveAction's arguments is key. We introduce the notion of *Semantic roles*; properties of entities that reflect their status in the relation. There are three semantic roles: $\mathit{Agent}(\cdot)$ (the property of being the entity doing the action), $\mathit{Patient}(\cdot)$ (the property of being the entity upon whom the action is being done), and $\mathit{Event}(\cdot)$ (the property of being the action entity). $\mathit{Agent}(\mathsf{x})$ means that the entity referred to by x has the semantic role of an Agent, and similarly for the other semantic roles. The TransitiveAction predicates ordering is that of Event, Agent then Patient.

The predicates $\mathit{Agent}(\mathsf{x})$, $\mathit{Patient}(\mathsf{y})$ and $\mathit{Event}(\mathsf{z})$ in the example above indicate that the dog is the agent (and thus *chasing*), while the cat is the patient and is being chased. $\mathit{Event}(\mathsf{z})$ indicates that $\mathit{Chase}(\mathsf{z})$ is the action entity.

Agents utilize a set of linguistic behaviors – a language – in order to communicate with each other. Their language transforms their internal representations – *meanings* – into an external *form*, comprised of words organized into sentences. Languages are bidirectional, *encoding* meanings as forms and *decoding* forms to meanings. In this work we assume all agents accurately perceive and utilize the same FOL internal representation.

Linguistic Behaviors

Linguistic behaviors specify how to encode and decode. Each linguistic behavior specifies a different encoding/decoding for some meaning/form. Agents will have conflicting linguistic behaviors that provide different encodings/decodings for the same meaning/form. The goal is for the population of agents to constrain their set of linguistic behaviors such that they can effectively communicate. The metric to evaluate successful communication is described in the "Evaluation Metrics" section.

In this work we will focus on *compositional languages*. A compositional language has structured forms where the meaning is a function of the meanings of the parts and the way they are combined (Johansson 2006; Krifka 1999).

Consider the meanings of the two sentences below:

```
The dog chases the cat.
The cat chases the dog.
```

Both sentences have the same constituent parts, however the ordering of the parts affects their meaning. Since English follows a Subject-Verb-Object ordering, the first element has the semantic role of "agent" and the third element that role of "Patient"

Notice that there is no word in the lexicon that specifies the semantic role of "dog". This meaning is specified in the structure of the sentence.

To implement a compositional language, agents will have to specify how meanings are encoded as words, and secondly how words are combined into sentences and how the structure of the sentence encodes meaning. These components of a language are called the *lexicon* and the *grammar* respectively.

We provide agents with a strategically simplified set of linguistic behaviors in which sentences will be limited to 3 words. As we already mentioned, all agents are utilizing the same internal representation of scenes. The environment will have a fixed, and known to all, set of entities.

Here is a simple language as an illustration:

- Lexicon:
- $w_1 \leftrightarrow \mathit{Dog}(\cdot)$
- $w_2 \leftrightarrow \mathit{Cat}(\cdot)$
- $w_3 \leftrightarrow \mathit{Chase}(\cdot)$
- $w_4 \leftrightarrow \mathit{Car}(\cdot)$
- $w_5 \leftrightarrow \mathit{Ball}(\cdot)$
- $w_6 \leftrightarrow \mathit{Jumps}(\cdot)$
- Grammar: Agent-Patient-Event

Figure 2: Sample Speakers Language

The lexicon in Figure 2 specifies how words are linked to meanings. There are only 6 words, $w_1 \dots w_6$. w_1 is mapped to the entity *Dog*$(\cdot)$, which means that w_1 refers to some entity (the argument) that has the property of being a Dog.

The ordering of elements in the grammar indicates the semantic roles of the referred entities. In this example, the entity referred to by the first word will be labeled *Agent*$(\cdot)$, the second entity will be labeled *Patient*$(\cdot)$, and the third entity will be labeled *Event*$(\cdot)$. Together, the lexicon and grammar provide a way of encoding (producing) and decoding (parsing).

Take the sentence $w_2 w_5 w_3$. First, each individual word is parsed and the meanings identified using the lexicon: $\mathit{Cat}(\mathsf{x}) \wedge \mathit{Ball}(\mathsf{y}) \wedge \mathit{Chase}(\mathsf{z})$. Then, taking the order of the words into account, we can identify the semantic roles of each entity, and then the relationship between all the entities.

To parse a sentence the process goes backwards. First the words for each entity in the scene are found based on the lexicon; then based on the semantic role of each entity in the scene the words are placed in the appropriate order.

Following (Vogt & Coumans 2003; Steels 1996) we implement the lexicon as an association matrix. For every word-meaning pair there is an association weight. When finding the meaning for a word the meaning with the highest weight is chosen. When producing a word for a meaning, the highest weighted meaning is chosen.

The grammar is implemented as an association matrix as well. When parsing or producing a sentence the grammar with the highest weight is chosen. To simplify notation we only indicate the winning word-meaning pairs and highest valued grammars in the figures.

Language Games

A language game is a way for an agent to gain knowledge of another agents linguistic behavior. The language game, and the associated update rule, define the behavior of the agent in the system.

Language games were initially developed by Steels in (Steels 1996). A (pair-wise) language game is an interaction between two agents, one of which is designated the speaker and the other the hearer. There are two main variants of the game, which we call the observation game and the guessing game following (Vogt & Coumans 2003). We will focus on the observation game.

In the observation game two agents establish joint attention on some part of the environment. This could be an object, or a complex scene describing several objects together. Regardless, the portion of the environment which is agreed upon will be called the *topic*.

The speaker agent produces a sentence that represents the topic. This sentence is passed to the hearer. The hearer determines if their language would produce the same sentence for the same meaning. If the hearer's language would do so, the game is successful. If not, the game is a failure.

In cases of failure, the speaker and hearer decrease the associations between the sentence and topic that they were using. In the case of success, the association between the sentence and the topic is increased, and competing associations are decreased (Vogt & Coumans 2003). The underlying principle of the update rules is for agents to change their language so that they are closer to each others language.

Evaluation Metrics

The emergence of a linguistic norm will allow agents to effectively communicate. We test the extent to which the norm is shared in the population by measuring the communicability of the population.

The communicability of two agents, one being the speaker and one the hearer, is the fraction of all possible scenes in the environment that the speaker can accurately communicate to the hearer. For each scene the encoding of the speaker is decoded by the hearer; if the hearers decoded meaning matches the speakers meaning communication was successful. The communicability over a population is the average communicability over all pairs of agents as speaker and hearer. Communicability will be a value between 0 and 1.

When the communicability of a population is 1, every pair of agents can accurately communicate all scenes. A linguistic norm has been achieved – all agents have constrained their linguistic behavior. An agent that violates this linguistic norm is sanctioned in the sense that it cannot accurately communicate.

With respect to norm emergence, we are interested in the time to emergence, which we measure in terms of the number of interactions before a norm emerges.

Imperfect Knowledge Leads to Ambiguity

The observation game defined in the "Language Games" section allows agents to gain knowledge of other agents linguistic behaviors. However, the knowledge gained through the game is imperfect. The imperfect knowledge leads to ambiguity on the hearers part of what the linguistic behaviors of the speaker are. We call this the *ambiguity problem*

Figure 3 illustrates the ambiguity problem. In it a shared scene, at the top, is described by the speaker. The speaker produces the utterance "gavagai mischal wooziar". The hearer has many possible interpretations of how the speaker produced the sentence that differ on what each word means.

The observation game only provides knowledge of the speakers sentence. Each word of the speakers sentence can refer to any of the three possible entities in the scene. All 6 possible interpretations are correct, in the sense that the speaker could be utilizing such a linguistic behavior. The hearer agent simply cannot determine the correct interpretation from the knowledge in the observation game.

As we saw above, the update rule for the hearer depends upon accurate knowledge of the speakers linguistic behavior. If the hearer does not choose the correct interpretation, then its update will lead to a decrease in communicability.

The interaction between the lexical component and the grammatical component in a hearer's language causes ambiguity in interpretation of the speaker's sentence. The hearer does not unambiguously know what the speaker's language is.

Strategies for Reducing Ambiguity

Techniques for reducing ambiguity revolve around two methods:

1. Constrain the complexity of the behaviors so that perfect knowledge can be achieved;
2. Provide extra knowledge to aid in discriminating the correct interpretation.

The first approach can be implemented as a "developmental trajectory". Initially the complexity of the linguistic behaviors would be constrained, i.e. agents will utilize languages with a fixed grammar. Once a high level of communicability has been reached, the constraint is lifted. (Lakkaraju, Gasser, & Swarup 2008) focuses on a developmental trajectory approach to language convergence.

The second option requires extra knowledge to be obtained from some method. We focus on this approach in this paper.

Text-Based Ambiguity Reduction Strategies

Our method for adding extra information is for agents to infer structural information through the use of *texts* that provide purely syntactic information about a language. Texts can be likened to *ambient language*. While texts carry no semantics (they are just ungrounded strings that need to be interpreted) the hearer can compare them with its world model to identify object-word correlations and word-position patterns, gaining ambiguity-reducing knowledge.

Texts have the advantage that they can be produced without the expensive co-presence and coordination of pointing and specific reference required for standard observation games. This is somewhat balanced, however, by the lack of semantic knowledge with which to ground the syntax present in the text.

We call our text-enhanced version of the standard observation game the *Text-Based Observation Game* (TOG). In the TOG, for each scene and interaction, the hearer receives a descriptive utterance and also has access to a text generated by the speaker. The hearer keeps track of the scenes it has encountered.

We assume three things, (1) that generated texts are about the environment; (2) the environment is stationary; and (3) all agents are using the same type of language (i.e. same positional type of grammar, with possibly different position-role assignments). Given these assumptions, an agent can statistically match the frequency of elements in the text to elements in the environment and gain an approximate understanding of the meanings of some of the words. We argue that this information is enough to resolve, to a large extent, the ambiguity problem.

An example illustrates the proposed technique. Suppose we have the speaker's language that is defined in Figure 2. We can generate a text for the speaker by having the speaker encode randomly drawn scenes from the world. One possible text is shown on the left hand side below:

1	$w_1 w_2 w_3$
2	$w_1 w_4 w_3$
3	$w_5 w_2 w_6$
4	$w_5 w_4 w_6$
5	$w_2 w_4 w_6$
6	$w_2 w_5 w_3$
7	$w_5 w_1 w_6$
8	$w_1 w_5 w_3$

	f	f_1	f_2	f_3
w_1	4	3	1	0
w_2	4	2	2	0
w_3	4	0	0	4
w_4	3	0	3	0
w_5	5	3	2	0
w_6	3	0	0	3

Since the agents use a positional grammar in which a words position in a sentence impacts its interpretation, the frequencies of the words in various positions provides knowledge about the words' functions in the sentence. From the text, we can calculate the frequencies of words and their position in the sentence. This frequency table is shown next to the text. The first column is the total frequency of the word in the text, and column f_i indicates the frequency of the word in position i of the sentence.

The frequency table provides insight into the syntax of the language. Namely, note that some words (such as w_1 and w_2) only appear in positions 1 and 2 throughout the text. On the other hand, some words (such as w_3) only appear in position 3 throughout the text.

The words in the text reflect patterns in the world. According to the speaker's language in Figure 2 w_3 refers to the predicate *Chase*$(\cdot)$. Every scene that contains this predicate identifies it as having the event semantic role. Thus, the

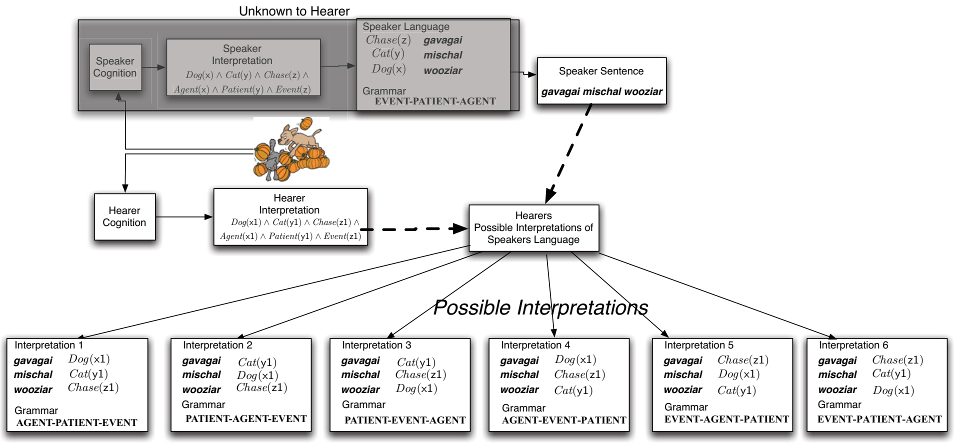

Figure 3: On the left is the shared scene that is viewed by both the hearer and speaker. The speakers language encodes the scene as "gavagai mischal wooziar". The hearer can interpret the sentence with reference to the shared scene in 6 different ways, or strategies. Only one of the interpretations matches the speakers language (shown in the shaded area). Each interpretation is a hypothesis about the speakers language based on the knowledge gained in the observation game. For example, Interpretation 1 hypothesizes that the speakers language encodes *Dog*$(\cdot)$ as gavagai, *Cat*$(\cdot)$ as mischal and *Chase*$(\cdot)$ as wooziar; and that the speakers grammar is of the form Agent-Patient-Event

speaker's language assigns the word to the third position in every sentence.

In addition, there are distributional patterns that can be elicited from the text. In addition to the text generated from the speaker, the hearer has a *scene store* that contains scenes that the hearer has witnessed. Similar to the statistics generated for a text, we can generate the frequency with which predicates appear with certain semantic roles.

Suppose the hearer agent has seen two scenes, where the frequency table is:

	f	f_{Agent}	$f_{Patient}$	f_{Event}
Dog(?)	2	2	0	0
Cat(?)	2	1	1	0
Chase(?)	2	0	0	2

This table indicates the frequencies of predicates from the scene store. The first column indicates the frequency of the predicate in the scenes, and the rest indicate the frequency of the predicate in a particular semantic role (i.e., as a patient, or event etc.)

Comparing the two frequency tables we can notice that some of the words match in distribution over positions with the distribution of the predicates over semantic roles. This correlation between the distribution of the word over position and the distribution of a predicate over roles provides evidence that word w_3 should be linked to the Chase predicate. With this information, we can weight the interpretations of the speaker's sentence and then choose the interpretation that matches the text the best.

The correlations offered by the texts do not fully specify the language of the speaker. However, they provide enough evidence, under the assumptions made above, to reduce the ambiguity in the interpretation of the outcome of a language game.

Using the texts agents develop *Language Hypothesis* – informed guesses about the structure of a language. In the text based observation game, the hearer constructs a language hypothesis through the evaluation of distributional similarity between words in the speakers text and its scene store. The language hypothesis will be used to weigh the hearers strategies.

Text Based Observation Games

The Text based observation game allows hearer agents to utilize texts to generate language hypotheses about the speakers language.

The hearer creates a set of strategies that define the possible structures of the speakers language. For every language game there are 6 possible strategies that the hearer has to choose from. Each strategy proposes a different updating of the speaker and hearer agents lexicon and grammar weightings.

Only one of the strategies matches the language of the speaker, this is called the *correct* strategy.

The hearer uses four metrics to evaluate and weight the strategies:

Lexical Metric: x_l Percentage of the word meaning pairs in the strategy that correspond to word meaning pairs in the hearers language.

Grammatical Metric: x_g Set to 1 if the grammar in the strategy is the same as the hearers current grammar.

Text Lexical Metric: x_{tl} Percentage of word meaning pairs in the strategy that are in the language hypothesis derived from the text.

Text Grammatical Metric: x_{tg} The weight of the strategies grammar in the language hypothesis derived from the text.

The hearer assigns a weight to each strategy based on the weighted sum of all four metrics:

$$w_l x_l + w_g x_g + w_{tl} x_{tl} + w_{tg} x_{tg}$$

where, $w_l + w_g + w_{tl} + w_{tg} = 1$.

w_l,w_g,w_{tl},w_{tg}reflect the preferences of the hearer on how to change its language. For instance, a weighting of $w_l = 1.0, w_g = 0.0, w_{tl} = 0.0, w_{tg} = 0.0$ means that the hearer agent prefers strategies that match its current lexicon, which means that they will produce the least amount of change to its lexicon. The strategy with the highest correspondence to the hearers language will be chosen in that case.

The exact method by which the agent updates its weights is given below. the highest weighted strategy is denoted by $\vec{\theta}$. The correct strategy is denoted by θ^*. We can also construct a "strategy" that reflects the way the hearer decoded the sentence, this is denoted as $\tilde{\theta}$.

An agent updates its association matrix according to:

Comprehensible & Success Method:

- Speaker increases θ^*
- Hearer increases $\vec{\theta}$

Comprehensible & Failure Method:

- Speaker decreases θ^*
- Hearer decreases $\vec{\theta}$
- Hearer decreases $\tilde{\theta}$

Incomprehensible & Failure Method:

- Hearer adds unknown words from $\vec{\theta}$to its language.
- Speaker decreases θ^*
- Hearer decreases $\vec{\theta}$

Incomprehensible & Success Can never happen!

Results

Our experiments indicate that Text based learning provides substantial extra information to allow for a significant reduction in ambiguity. This lead to a 40% decrease in the amount of time to converge as compared to a method that does not use texts.

This performance increase was due to a decrease in the number of mistakes on the hearer's part when choosing a strategy. The text based strategy had nearly half the errors of the non text based strategy.

We performed 3 different experiments. Each experiment used an environment with 30 entities and a population of 20 agents. Three of the 30 entities were events, and the other 27 were objects. There were $27 * 26 * 3 = 2106$ possible scenes described in the world. The number of words was not bounded, but agents could only construct 3 word sentences.

The three semantic roles were fixed for all agents. In all experiments all the agents started out with random grammars and empty lexicons.

There are 5 important parameters. $n_a$is the number of agents in a population, $n_o$is the number of object meanings, $n_e$is the number of event meanings, $n_t$is the size of the text, $n_m$is the size of the scene store for each agent.

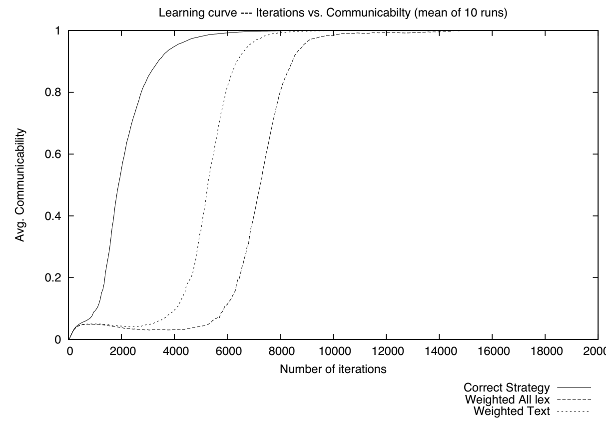

Figure 4: Convergence curve for 20 agents and 30 meanings.

For the experiments here, the parameters were: $n_a = 20$, $n_o = 27$, $n_e = 3$, $n_t = 100$, and $n_m = 20$

The experiments were:

Correct Strategy The perfect knowledge situation. The hearer agent was given the correct interpretation.

All Lex In this experiment we measured how fast agents could converge when they relied only on x_l. Correspondingly, we set $w_l = 1.0$ and everything else to 0.0. This is the imperfect knowledge case.

Weighted Text In this experiment we measured how fast agents could converge when they relied on a text. The agents used a text, and the weights were set to: $w_l = .5, w_{tl} = 0.5$ and all other weights set to 0.0.

Figure 4 shows the convergence curves (averaged over 10 runs) for these different experiments.

In an optimal setting there would be no ambiguity for the hearer agent in determining the language of the speaker — thus leading to quick convergence. We tested this hypothesis by simulating a situation where the hearer agent always has access to the correct strategy. Figure 4 shows the convergence curve for the three experiments. A convergence curve plots the average communicability over all pairs of agents in the population over time. Communicability can range from 0, where none of the agents speak the same language, to 1, where all the agents speak the same language.

The solid line in Figure 4 is the convergence curve for the correct strategy experiment. It is clear that when the correct strategy was given to the hearer agent, the population converged quickly to a common language. The quick increase in communicability corresponded to the point when all the agents in the population converged on the same grammar.

On the other hand, the dashed line (far right in Figure 4) shows the convergence curve for the All Lex experiment. We can see that in this case there is a substantial increase in the number of interactions for agents to converge on a single

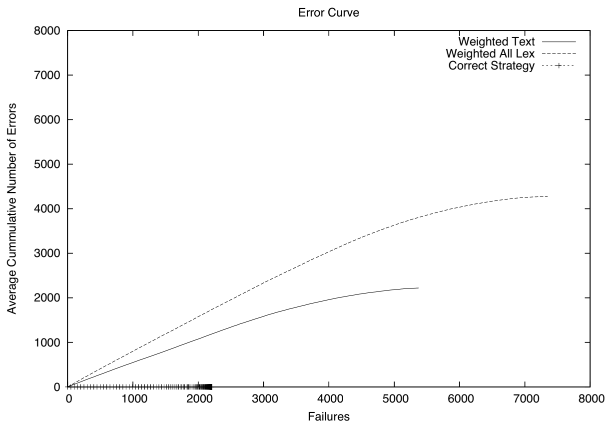

Figure 5: We see a steady increase in failures, then no failures occurring. The number of failures is longer for the all text case and the text case as compared to the correct strategy case.

language. The population does converge, yet it takes a much longer time to do so.

The dotted line between the correct strategy convergence curve and the all lex convergence curve is the convergence curve for the text experiment. While the population does not converge as quickly as in the correct strategy case, the population does converge faster than in the All Lex experiment.

As we noted above, the decrease in the amount of time to convergence is due to the lessening of ambiguity in the hearer agent. The text provides information about the speaker's language that allows the hearer to make a better choice of strategy. The increase in knowledge gained by texts translates into a decrease in the number of wrong strategies chosen by the hearer agent.

The *error rate* of an agent is the number of times it picks the wrong strategy as a hearer. Figure 5 shows the failure error rate — the rate of error on unsuccessful language games. We can see that the All Lex experiment had the most number of errors, whereas the Weighted Text experiment had far fewer. The correct strategy experiment did not have any errors, as the correct strategy was given to the hearer.

The slope of the error curve indicates the errors per failure. The optimum is a slope of 0, where there is no error per failure. The worst is a slope of 1 where there is an error on every failure. We can see that the slope of the Weighted Text experiment is much closer to 0 than the slope of the Weighted All Lex experiment, which indicates a significantly smaller error per failure rate.

Conclusions & Future Work

Our objective is to understand norm emergence in complex settings. In this paper we use emergent language as a model distributed information system that can be used to study aspects of norm emergence in complex settings.

We showed how ambiguity due to imperfect knowledge affects the amount of time it takes for a norm to emerge. We outlined two solutions to this problem and provided a detailed solution in the form of *Text Observation Games*. Through the use of collections of sentences generated at random from the speakers language – a text – we show how to handle the ambiguity that arises. Through the use of texts, there is a significant decrease in the time to establish a linguistic norm, as well as a significant decrease in the number of errors.

Numerous further extensions to this work are possible. The current interaction topology allows all agents to interact with all other agents, which is clearly unrealistic. We intend to explore and compare our methods on scale-free, small world and other networks; (Savarimuthu *et al.* 2007) has shown norm emergence with similar topologies. Currently texts are generated per interaction, making them accurate reflections of the speakers language. A more realistic approach would have texts generated only periodically while. We plan on investigating how the time between text generation can influence norm emergence.

References

Andrighetto, G.; Campenní, M.; Conte, R.; and Paolucci, M. 2007. On the immergence of norms: a normative agent architecture. In *Proceedings of AAAI Symposium, Social and Organizational Aspects of Intelligence, Washington DC.*

Johansson, S. 2006. Working backwards from modern language to proto-grammar. In *Proceedings of the 6th International Conference on the Evolution of Language*, 160–167.

Krifka, M. 1999. Compositionality. In *MIT Encyclopedia of Cognitive Science*. MIT Press.

Lakkaraju, K.; Gasser, L.; and Swarup, S. 2008. Language scaffolding as a condition for growth in linguistic complexity. In *In Proceedings of EvoLang 2008*.

Opp, K.-D. 2001. How do norms emerge? an outline of a theory. *Mind and Society* 3:101–128.

Savarimuthu, B. R.; Cranefield, S.; Purvis, M.; and Purvis, M. 2007. Role model based mechanism for norm emergence in artificial agent societies. In *Proceedings of the Workshop on Coordination, Organizations, Institutions and Norms at AAMAS 2007*.

Shoham, Y., and Tennenholtz, M. 1997. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence* 94(1-2):139–166.

Steels, L. 1996. Emergent adaptive lexicons. In *Proceedings of the Fourth International Conference on Simulating Adaptive Behavior*.

Vogt, P., and Coumans, H. 2003. Investigating social interaction strategies for bootstrapping lexicon development. *Journal of Artificial Societies and Social Simulation* 6.