Imitation is not enough for lexicon learning

Jason Fleischer
The Neurosciences Institute
10640 John Jay Hopkins Drive
San Diego, CA 92121 USA
fleischer@nsi.edu

Jonathan Shapiro
Department of Computer Science
University of Manchester
Manchester M13 9PL UK
jls@cs.man.ac.uk

Abstract

Lexicon learning systems need to be concerned with more than just producing symbol usage agreement between agents, which is easy to acquire through imitation. Lexicon learners should also explicitly attempt to increase the mutual information between their symbol usages (a measure of the usefulness of the symbols for transferring information between agents). This paper argues that, although many lexicon learning algorithms presented in the literature do attempt to create highly informative symbol usages implicitly, there are good reasons to make the mutual information of symbol usages an explicit goal of the lexicon learning system. Some first steps in this direction are provided in this paper. It presents lexicon learning experiments using both purely imitative and explicitly information maximizing algorithms. The results of these experiments are used to support the thesis of this paper, that lexicon learning algorithms should explicitly attempt to produce high mutual information symbol usages.

1. Introduction

Lexicon learning is a task where a group of agents learn to use symbols to label meanings. The goal of the task is to enable agents to transmit meanings to each other using the symbols contained in the lexicon. Therefore, a lexicon could not be called successful, in any sense, unless a signal-receiving agent could reliably come up with the same meaning that the signal-transmitting agent intended.

This interpretation of lexicon learning has significantly influenced the way it has been approached in the literature. The primary focus of recent research has been on how the agents can use symbols to denote meanings *coherently* (i.e., that meanings can be reliably transmitted). However, there has been little focus on another important aspect of lexicon learning: that the goal of communication is to transmit information. As such, it is little use for the agents to have a coherent lexicon if they cannot distinguish between meanings that are useful for task-accomplishing behavior or survival. For instance, a degenerate case of enforcing coherence in the lexicon is simply to label every meaning with the same symbol; such a lexicon would be of no use in trying to transfer information from one agent to another.

Much of the original work in lexicon learning asked the question, “Under what circumstances will agents evolve an explicit signaling system?” See, for instance, the work of (Werner and Dyer, 1992; MacLennan, 1992). More recently, a number of approaches have looked at how lexicon formation might occur as a result of agent’s imitating each other’s symbol usages (e.g., Steels, 1998; Batali, 1998). These approaches have focused on the contribution of imitation to the learning process, and the dynamics of lexicon formation. However, they do not directly address the problem of learning an informative lexicon, but instead typically have “hidden” features that help create informative symbol usages. These safeguards are neither theoretically motivated nor do they have any performance guarantees.

In contrast, we are interested in the use of lexicon learning among robots to enable them to communicate with each other about a task (Fleischer, 2004). In the course of this work, an imitative lexicon learning algorithm was implemented and found to be susceptible to the degenerate case of labeling every meaning with the same symbol. The reason for this problem was two-fold: (1) the robot data is noisy and does not lend itself to easy extraction of meanings, and (2) most imitative lexicon learning algorithms contain some hidden features that ensure the creation of an informative symbol usage, whereas the algorithm implemented above did not. This result exposed an aspect of imitative symbol learning that is not explicitly addressed in the literature: imitation cannot guarantee an informative lexicon in the absence of other mechanisms.

This paper addresses the problem by calling for research into lexicon learning methods that explicitly address the problem of learning informative symbol usages, and by investigating one such algorithm.

Section two of this paper argues that there are benefits to be had by making informative symbol usages an explicit goal of the lexicon learning system. First, a definition of the symbol usage informativeness is presented, along reasons why an informative symbol usage is desirable for a set of communicating agents. Next, two of the more known examples of imitation-based lexicon learning algorithms from the literature are presented, and it's illustrates how they contain implicit or hidden safeguards that cause these algorithms to create informative symbol usage.

In order to show the potential problems of imitation learning, section three presents an imitation-based symbol learning algorithm that lacks the implicit safeguards of the approaches in the literature. Experiments demonstrate how it performs in both simulated lexicon learning tasks and also in a location-naming task on a mobile robot. The simulation results give an indication of both ideal performance and what types of meaning confusion will most adversely affect the information of the learned symbol usages. The robot lexicon learning task is an example of what can go wrong with imitation learning when there is a lot of perceptual noise.

Section four makes a first stab at a learning system that explicitly seeks more informative symbol usages. The learning system is based on negotiation; when agents disagree about how to label a meaning, they enter negotiations where each proposes a labeling that will increase its estimate of symbol usage information entropy. Agents only accept a proposal that does not decrease their own estimates of information entropy. Simulation results show that this learning system can, in the right circumstances, produce more informative symbol usages than the imitation algorithm. In terms of absolute performance, however, imitation can always be tuned to produce more informative symbol usages. An analysis of this problem is offered.

The conclusions section ties together the strands of argument about why information maximization should be an explicit goal of lexicon learning. Finally, future directions for producing information maximizing algorithms are discussed.

2. Background

2.1 Measuring information

Information theory provides tools useful both for evaluating how informative a set of symbol usages is, and also for providing a measure which can be maximized explicitly during learning to ensure that the symbol usages are informative. The concepts of information theory were introduced by Shannon (1948), and a standard textbook in the area is (Cover and Thomas, 1991).

Formally, an information channel is a random variable X, that can take a discrete and finite set of values (or signals) $\mathcal{X} = \{x_1, x_2, \ldots x_n\}$ that occur with probabilities $p(x_1), \ldots, p(x_n)$, such that $p(x_i) \geq 0 \ \forall i \in \mathcal{X}$ and $\sum_{i=1}^{n} p(X_i) = 1$, forming the discrete distribution $P(X)$. Then the entropy, H, of the channel is defined as

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i).$$

which is, in fact, the average information content of the channel. Note that the maximum of this function occurs when all the signals are equiprobable. That is, the maximum information content of communication will take place when all the symbols are equally likely to occur. Also, information entropy will be zero when the probability of a single symbol is one, and all the others are zero. This is the degenerate case where every meaning is labeled by the same symbol.

Mutual information (also called information gain) is the amount of information that we gain about one random variable if we know the value of another. It can be defined as,

$$H(X;Y) = H(X) + H(Y) - H(X,Y).$$

Mutual information is a useful measure of the information content of the symbol usage between two agents. In a test set of meanings, each agent generates a set of symbols to describe those meanings. Each agent's description of the test set is a random variable, which we'll call S_a for the symbols used by agent a, and S_b for agent b. The mutual information between the symbol usages is then $H(S_a;S_b)$. Note that $H(S_a;S_b)$ can never be greater than $\min(H(S_a), H(S_b))$. Also, $H(S_a;S_b) = 0$ only if the two variables are independent, and is equal to the maximum only if the two variables are maximally correlated. So this measure has a maximum and minimum, and progresses in a fashion that is natural to the quality we want to describe.

The mutual information between symbol usages in a test set of meanings is used in this paper as a means to evaluate how informative the symbol usages learned by the agents are. However, this quantity requires knowledge of the output of all the agents on the entire set of inputs. When an agent is employing a learning algorithm that explicitly tries to maximize information it will be unlikely to have access to such global performance knowledge. Instead, the agent can assume that by increasing the information entropy of its own symbol usages, something which is much easier to estimate, that it will increase the mutual information between symbol usages. This method is discussed further in section 4.1, which presents the explicit algorithm.

2.2 Why maximize information?

The research in lexicon learning to date has predominantly focused on the methods by which agents learn and the properties of learning such lexicons. Much work has also taken the approach of trying to understand how natural language evolved and how humans create meaning. If the purpose of investigating lexicon learning is to better understand lexicon formation and language evolution in humans, then explicit information maximization algorithms should be of interest. It seems unlikely that imitation is the only mechanism involved in lexicon acquisition and language evolution; there could be specific features of these systems whose function it is to maximize the informativeness of the symbols used.

In contrast, one also might be interested in having agents learn a lexicon in order to *accomplish a task*. Agents that work together in a group can increase their task performance by using some form of communication (Balch and Arkin, 1994). One can obviously design and implement a communication system for such agents, but there are possible benefits to a communication system that the agents adapt for their own uses. These are the same arguments that are normally made for adaptive behavior in general: more robustness, the ability to react to changing circumstances, the ability to tackle problems that the agents were not explicitly designed to solve.

If the purpose of lexicon learning is to enable agents to communicate about objects in the world (e.g., robots describing navigation routes to each other by naming landmarks along the route), then it becomes very important that the symbols used should be useful for the task. Defining the usefulness of a symbol system for a task is a difficult thing, that perhaps can only be assayed on a case-by-case basis. However, it is impossible for a symbol system to be useful if it is not informative. Take the example of a lexicon that names landmarks experienced by mobile robots. If almost every landmark is called by the same name, then a route description generated using that lexicon would be of very little navigational use to a robot trying to follow the route.

When task accomplishment is important, then it seems obvious that informative symbol usages are an important goal. When mechanisms for maximizing information in the lexicon are implicit, such as in the methods presented in the next section, then it can be difficult to ensure that the results will be as desired. The best way to make sure that symbol usages remain informative is to make it an explicit goal of the learning system.

2.3 Literature

This section presents two well-known examples of imitative lexicon learners that do not explicitly try to maintain the information content of symbol usages. However, since both are able to produce lexicons with high mutual information they must have some implicit mechanisms that keep them from devolving to the solution of naming all meanings with the same symbol. The following paragraphs illustrate how these hidden mechanisms work.

Steels (1996) introduced naming games as a method of lexicon learning. A naming game is a game in that sense that agents try to "win" by successfully communicating with each other. If both agents understand a symbol to represent the same meaning then a naming game is successful. Losses, i.e., communication failures where the agents do not use a symbol to describe the same meaning, give the agents data they use to do better at the game in the future. Steels and colleagues have introduced a variety of "language games" that perform lexicon learning[1]. Although the language games have different algorithms, they can all be described by the same general framework:

> Two agents are selected at random from the population. One agent is randomly selected as the speaker, and the other as the listener. Both agents are presented with a context containing several objects. The speaker selects a topic object from the context, and produces an utterance to describe it. The speaker chooses the most "successful" symbol in its repertoire to represent the topic object, or makes one up if it doesn't already know one for that object. Both agents increment counters of symbol usage and naming game success, whose values they use to determine which symbol to generate when speaking.

Note that the naming game is imitative because the agents use their estimations of symbol occurrence and success rate to decide which symbols to use. That is, by observing the usage and communicative success of the rest of the population, they determine their own symbol usage.

One particularly important element of the language games, for the purposes of this discussion, is that the algorithm makes up a new word (i.e., symbol) randomly to describe any new object. The likelihood of the population as a whole using the same symbol to describe multiple meanings (and thereby loosing information in its symbol usages) is directly related to how an agent randomly chooses a new word to describe an object. Imagine two agents, each encountering different objects for the first time. They now (separately) make up words for those objects and use them in naming games with other agents. If both agents are fairly likely to make up the same word, then its possible for the population as a whole to adopt that word to represent two different, and perfectly distinguishable, objects. Unfortunately, the mechanism used in the naming games to make up new words is never completely explained in the literature, so it the true probability of this circumstance occurring is not known.

[1]The naming game, the observational game, the guessing game, the selfish game. These are explained and reviewed in (Vogt, 2000) and (Steels, 1998).

Note that if selection is equiprobable and the number of words in the possible set is orders of magnitude greater than the number of objects multiplied by the number of agents, then there is very little chance of symbol usages having low information. This can be seen from the following argument. (1) Since agents learn from the symbol production of the rest of the group, it is not possible for the final symbol assignment for a meaning to be anything other than one of the symbols made up by an agent first encountering that meaning. (2) It is therefore unlikely that one symbol will be used for multiple meanings if there are many more symbols than possible occurrences of an agent first experiencing a meaning. Yet some of the naming games reported in the literature do have symbol usages where more than one meaning generates the same symbol (see the "specificity" measurements in Vogt, 2000). This naturally raises the question of how tunable this information loss is, and whether it can effectively be prevented if it is not wanted.

Another well-known approach to imitative lexicon learning is that presented by Batali (1998). In this simulation, the agents learn a lexicon where the symbols are composed of morphological components, in the same way that words are built out of letters. The agents themselves are recurrent neural networks that, when presented with a meaning vector as input, produce strings of symbols as outputs. The strings are truncated at 20 characters (even if the recurrent network would produce more), and each character can be one of four letters.

Learning is directly imitative. An agent acting as the hearer trains its network to produce the same meaning encoded by the speaker. The same network that interprets symbols to meanings is also used to send symbols when the agent acts as speaker, so training as the hearer affects production as speaker.

Note that the set of potential strings (the functional equivalent of symbols) is of size 4^{20}. Since there are only 100 possible meanings we are once again confronted with a case where it is very unlikely that random initializations would produce the same symbol being assigned to different meanings. However, it is more complex than the case presented by the naming games of Steels. The associations between meaning and symbol are produced by a recurrent network. Therefore, the relationship between input and output is much more complex than the case of simply counting occurrence and success rates. Over the course of training, it seems inevitable that the network weights will become modified so that the agent will produce symbols that it neither originally produced after initialization nor was trained to produce. In this case, the question of how to control the information content of symbol usage has no obvious solution.

3. Imitation learning

This section demonstrates that an imitation based algorithm without the hidden safeguards can develop very uninformative symbol usages. This provides a proof of the dangers of relying only on imitation, which motivates the development of mechanisms that explicitly maximize information in the lexicon.

3.1 Algorithm

At this point it is important to distinguish between a meaning and a perception. In this paper, a meaning is a category or object that has an identity independent of agent perceptions. That is, it is a real-world location or it is an abstract category in a simulation that the agents are trying to communicate about. In some of the experiments, agents will have different perceptions of these meanings. For instance, they may be unable to differentiate between two meanings. In the robot experiments the agents self-organize categorizations, thereby creating arbitrary meanings from unlabeled data that will be different from agent to agent.

The lexicon learning system is based on each agent maintaining estimates of the conditional probability that it should generate a particular symbol, given that it has perceived a particular meaning. When an agent is required to generate a symbol to represent a perception, it generates the maximum likelihood symbol (the symbol with the highest conditional probability given the perception experienced).

The set of possible perceptions and symbols is fixed before lexicon learning occurs. For instance, in a simulation there may be ten meanings to be learned. These meanings will not change identity or grow in number during the simulation. Likewise, the agents might be given five arbitrary symbols to label these meanings at the beginning of lexicon learning, and the symbols will not change identity or grow in number during the learning process.

The conditional probabilities are stored in a single-layer neural network, with no bias input, using the softmax activation function. The input to the neural network is a 1-of-n encoded vector of the current meaning (as perceived by the agent — recall that agents may have different perceptual abilities). The activations of the output nodes are the conditional probabilities given the perception on the input nodes. Note that a single-layer network is sufficiently powerful to encode the probabilities because the inputs are mutually exclusive and therefore orthogonal in the input space. The weights of every agent's neural network are initialized using a zero-mean normal distribution.

The agents then proceed to learn a lexicon by imitating each other's usage. The total training data is partitioned into training, validation, and test sets of equal

no error	disagreements	0 (0)
	$H(C_a; C_b)$	3.29 (0.01)
	$H(S_b; S_b)$	2.55 (0.05)
	unique symbols	6.57 (0.20)
systematic error	disagreements	28.2 (1.8)
	$H(C_a; C_b)$	3.26 (0.01)
	$H(S_a; S_b)$	2.21 (0.05)
	unique symbols	6.53 (0.18)
random error	disagreements	18.2 (0.7)
	$H(C_a; C_b)$	2.53 (0.04)
	$H(S_a; S_b)$	1.93 (0.04)
	unique symbols	6.40 (0.15)

Table 1: The mean (std. err.) performance over 30 trials for various perceptual error models in the simulation experiments. See section 3.1 for an explanation of the models and the measurements. The results are over a test set of 200 randomly drawn meanings.

sizes (300 each for simulation, 640 each for the robot data). The training set is re-ordered randomly at the beginning of every epoch. Training occurs on-line according to the following method:

> Each agent perceives a meaning in the training set according to its abilities, and encodes it as an input vector for it's neural network. Each agent then generates the symbol that has the maximum likelihood on the outputs, and transmits it to the other agent. When an agent receives the other one's output it turns the symbol into a 1-of-n encoded vector which is used as a training pattern for its own neural network. The network is trained using on-line gradient descent on the cross-entropy error function, with a learning rate of 0.01 and no momentum.

At the end of an epoch of training the agents are evaluated on a validation set. An error is counted for every meaning in the validation set which the agents generate two different symbols to describe. Training continues until a maximum number of epochs is reached (1000 for simulation, 10000 for robot experiments), validation error is zero, or validation error has not decreased for a certain number of epochs (500 in both cases).

3.2 Simulation experiment

In this simulation the agents were given ten symbols to label ten meanings. These ten meanings occur with equal frequency, but sometimes one or both of the agents may systematically or randomly confuse meanings with each other. In other words, the agents may not perceive the same meaning, and therefore the imitative symbol learning they perform will be subject to noise in the target values.

There are two ways the agents can perceive the ten meanings:

No error Both agents perceive perfectly.

Systematic error Agent a perceives meaning #1 as meaning #2 50% of the time and agent b perceives meaning #3 as meaning #2 50% of the time.

Random error Agent a confuses all meanings with $p = 0.1$. When confused it picks any other of the nine meanings with uniform probability. Agent b perceives perfectly.

In both systematic and random error models roughly 10% of meanings will be confused by one or another of the agents. Thirty simulations were run of each model, with different randomly initialized agents in each simulation.

The results of these simulations are summarized in table 1. Four different measures of agent performance are presented here, giving information on how much agreement there is on symbol usage, on how much information it is possible to transmit, how much information was transmitted, and how many symbols were used. The number of disagreements is the number of times the agents produced different symbols for the 300 randomly drawn meanings (w/ uniform probability) in the test set. $H(C_a; C_b)$ is the mutual information between the agent's *perceptions* of the meanings, i.e., the meanings with the perceptual error model applied. This is the maximum possible value that the symbol usage mutual information, $H(S_a; S_b)$ could ever hope to achieve. Finally, the number of unique symbols in the test set is also recorded, telling us how many of the ten possible symbols were used in the test set.

The empirical results show us that the imitative learning algorithm will produce a symbol usage that contains on average 77% of the information contained in the meanings when the number of symbols equals the number of meaning. It should be noted this loss of information is a "best-case" scenario. If the meanings are not equiprobable then it can only become worse.

Looking at the effect of the perceptual error shows firstly that both sorts of errors significantly reduce the amount of information in the learned symbol usages. What is interesting to note, however, is that in the case of the random error, the loss in $H(S_a; S_b)$ probably comes directly from the loss in $H(C_a; C_b)$ from the perceptual errors. In the random error model the symbol usage mutual information is 76% of the perception mutual information, a close match to the 77% found with perfect perceptions. However, systematic errors seems to cause a loss of symbol mutual information that is not accounted for by the corresponding loss of perception mutual information. In the systematic error experiments the symbol usage obtains only 68% of the mutual information of the perceptions.

measurement	mean	std err
disagreement rate	26.3%	0.4%
$H(C_a;C_b)$	2.20	0.02
$H(S_b;S_b)$	1.38	0.41
unique symbols	3.13	0.12

Table 2: Test set results for the robot landmark lexicon learning experiment (30 trials, 20 symbol lexicon), where each agent uses its own 20 node SOM to categorize landmarks. See section 3.3 for an explanation the setup.

These results tell us three important things. First, we shouldn't expect an imitative system without safeguards to learn symbol usages with as much mutual information as exist in the agent's perceptions. Secondly, any sort of perceptual confusion will only aggravate this problem. Thirdly, for a given level of loss in perceptual ability, a systematic error would seem to cause a greater loss of mutual information as a percentage of the possible information in the meanings. Other experiments that are not reported here with different numbers of meanings, symbols, and noise levels all support these conclusions.

3.3 Robot experiment

This section presents an experiment using the imitative symbol learning algorithm to try and generate a lexicon for use in a robotic navigation task. The lexicon consists of names for landmark types (in a human sense, something akin to "T-junction" rather than a name for one specific location like "T-junction #42"). The interested reader can find further details on both landmark lexicon learning and route description and navigation for this task in (Fleischer, 2004).

In fact, this experiment is originally responsible for our investigation of the topic of explicit information maximization in lexicon learning. The low information symbol usages learnt by the imitation algorithm spurred the need to develop a better method of lexicon learning.

In this experiment, two agents with different random initializations are trained on the perceptions of a single physical robot. Therefore, there is never any doubt about what the "meaning" is — both agents have the same raw sensor input. However, the sensor input is not a meaning in and of itself, but rather a high-dimensional, noisy, representation of the local area around the robot[2]. Each agent produces a fixed set of meanings for itself by training a Self-Organizing Map (SOM) (Kohonen, 1993) on a batch of representative landmark perceptions. The agents then categorize raw sensor data during lexicon learning by using the identity of the highest activation node of the map for a given sensory input as the agent's perception of that landmark. In other words, each agent will have similar but not identical perceptions of landmarks. Thus, we can expect both systematic and random perceptual error during lexicon learning.

The agents trained 20 node (4x5 toroidal topology) SOM networks on robot data representing 200 meters of travel in three different office corridors. They then used a a different 110 meters of data, taken from a subset of the same corridors, to train their lexicons. The lexicon learning data was randomly partitioned into training (1/2), test (1/4), and validation (1/4) sets. Training continued to a maximum of 10,000 epochs, but otherwise learning parameters are identical to those of the simulation. The agents were given a 20 symbol lexicon to learn.

Lexicon learning was repeated according to this setup 30 times with different agent initializations and different training data splits each time. The results can be seen in table 2, and they are quite disappointing. The agents produced a symbol usage with only 62% of the possible mutual information[3]. In light of the simulation results presented in section 3.2, it is clear that the agents must be suffering from extensive perceptual noise. This is, in fact, the case. By looking at any given pair of SOMs and sensory data sets, clear cases of systematic errors can easily be found. Also, when the two agents are given identical SOMs (i.e., they now have identical perceptions of the landmarks), they can learn to produce the normal (for the case of equal numbers of symbols and meanings) 76% of the possible mutual information.

These disappointing results highlight the importance of maintaining mutual information in the symbol usages, if the goal of lexicon learning is to produce a communication system that is useful for transmitting information between agents.

4. Negotiation learning

This section describes experiments using an algorithm that attempts to explicitly increase the mutual information of symbol usages. The agents do this by negotiating with each other about what symbol they should use for a particular meaning.

The motivation of using negotiation comes from the desire to produce explicit maximization of symbol informativeness without trying to compute things the agent will not typically have access to, such as the probability of a particular perception occurring (difficult in different environments) or symbol usage probabilities of all agents in the communication group (difficult when there are many agents). Although the scheme is not motivated by a particular cognitive or psychological model, it seems that explicit negotiation does take place when

[2] In actuality, a 20x20 occupancy grid updated with the data of the last 10 inches of travel, covering an area of 100x100 inches around the robot.

[3] Note that information entropy is a logarithmic scale, so seemingly small percentages of change can be very important in terms of system operation. In fact, the robots were essentially unable to navigate using this type of landmark description lexicon.

adult humans play games requiring the development of coordinated descriptions of spatial events (Garrod and Doherty, 1994), although in their case negotiation seems to function as a method of selecting which of several possible alternative description schemes to use in a particular game, rather than which scheme to use "for all time" as we're investigating here.

4.1 Algorithm

The negotiating agents are substantially similar to the imitation agents described in section 3.1. They generate symbols using the same type of neural network, and are trained using the same parameters.

However, a negotiating agent also maintains an estimate of the occurrence rate of symbols. It does this by counting the number of times it uses each symbol during a single training epoch. Before training begins each agent runs through the training set of meanings by itself in order to generate an initial estimate of the occurrence rates. At the end of an epoch, the agent replaces its current estimated symbol probabilities with the occurrence rates from the last epoch. Let us denote agent a's current estimate of the occurrence rate of symbol s as $P_a(s)$. Note that

These probability estimates are used by the agent to try and maximize the information entropy of its own symbol usage, which in turn makes it possible for the mutual information of symbol usages to become higher. Entropy is at a maximum when the symbols are equiprobable; equiprobable symbol use is therefore the direct goal of the negotiating agent.

The negotiation protocol is as follows:

1. Each agent, $\{a, b\}$, receives the meaning, perceives it according to its ability, and produces a symbol, $\{s_a, s_b\}$ using the method described in section 3.1.

2. If $s_a = s_b$ then train the networks using that symbol and return to step 1.

3. Otherwise each agent makes an ordered list of all symbols that occur equally or less often (according to its estimation) than the symbol it produced in step 1. The least used symbols are first in the list.

4. Each agent proposes a symbol (agent a proposes s_a^{prop}) from the top of its list and sends this symbol to the other agent.

5. Each agent calculates its willingness to accept the other's proposal. For agent a this value is

$$r_a = log \frac{P_a(s_a)}{P_a(s_b^{prop})}$$

6. If at least one of the signals is non-negative, then the agent with the largest willingness (r) accepts the other agent's proposal. In case of a tie, a winner is chosen randomly. Each agent trains its network using the winning proposal as the target. Return to step 1.

7. Otherwise, each agent removes the symbol from the top of its list. If both agent's lists are now empty they return to step 1. If at least one agent still has proposals left to make they return to step 4.

Obviously, coherent symbol assignment (where the agents agree on the symbols to use for all meanings) is a fixed point of this protocol. It's also the case that it can be shown that this protocol will always produce a coherent symbol assignment provided that (1) the agents perceive the meanings in exactly the same way (no disagreement from perceptual confusion of meanings), and (2) that the learning system, in this case the neural network, is capable of making the assignments proposed by the protocol (no disagreement because the meaning-symbol mapping dictated by the algorithm cannot be learned).

4.1.1 Proof of convergence

Theorem: Given a set of input patterns, and a pair of agents with different symbol assignments for those patterns, there is at least one pattern for which a symbol assignment is accepted and agreed upon by the agents.

Proof by contradiction: Assume that there are two different symbol assignments such that *no* symbol proposal made by one of the agents is accepted by the other. Consider the set of symbols that agent a uses more than any other symbols,

$$S_a^0 = \{s \,|\, P_a(s) \geq P_a(s') \,\forall s' \in S\},$$

where S is the set of all symbols. For any pattern which agent a assigns to a symbol in this set, any proposed symbol would be accepted by a. Thus, since we assume no proposal is accepted, the two agents must already agree on the patterns assigned to these symbols. Now consider the symbols used by agent a which are used more than any other symbols except those in S_a^0,

$$S_a^1 = \{s \in S \setminus S_a^0 \,|\, P_a(s) \geq P_a(s') \,\forall s' \in S \setminus S_a^0\}.$$

Note that agent b cannot propose a symbol in S_a^0 for any pattern that a assigns to a symbol in S_a^1 because a and b agree on those symbols and therefore must have the same estimate of occurrence rate. So a will accept any symbol proposed by b, and since we are assuming that no proposal will be accepted, it must be the case that for all patterns which a assigns to a symbol in S_a^1, b must assign the same pattern to the same symbol. Likewise, we build sets $S_a^2, S_a^3, \ldots S_a^i$ (where i is the number of such sub-sets in S) and show that the two agents must agree on the symbol-pattern mappings in those sets. The same argument can be used from agent b's standpoint. But then we have shown that the agents agree on symbols for all patterns, and have contradicted the assumption that they have not.

Corollary: If the learning systems can learn to assign any symbol to any pattern without affecting the mappings from the other patterns, this protocol will converge to a coherent set of symbol assignments.

The single-layer neural network in question will produce symbol usage coherence because the input patterns are orthogonal and there is no bias term in the outputs to link the symbol mappings of the patterns.

However, this algorithm does not converge to the optimal (maximum possible entropy) solution. This happens for two reasons. First of all, agreement is a fixed point in the system. So as soon as the agents agree on how to label a particular meaning, they will no longer be able to change that, even if some other assignment they make later in learning would mean that mutual information would be increased by using a different symbol. Secondly, an agent is not actually increasing the information entropy of its symbol usage, but is employing a heuristic approximation to direct maximization. The acceptance criteria will only increase information of symbol usage if the change in symbol occurrence rates after the new assignment is small and of equal size for all symbols. There are many times when this is might not be the case, e.g. if the agent assigns a new symbol to a meaning that occurs 70% of the time.

4.2 Simulation experiment

The simulations presented here are identical in procedure to the no noise simulations presented in section 3.2. They demonstrate that the negotiation algorithm *can* outperform the imitation algorithm it was designed to replace in certain circumstances, but that the imitation algorithm can be tuned to produce a more informative lexicon more reliably.

In this experiment 5, 10, 15, and 20 symbols were used by the agents to label the same twenty meanings. Thirty trials were made of each using different random initial network weights but the same no-noise data set. Figure 1 shows the mean values and three standard error bars of these experiments. It can be seen that when only five symbols are available, the negotiation algorithm is a clear winner. However, imitation is as good as negotiation when there are as many symbols as meanings. Once there are more symbols than meanings imitation becomes the clear winner. This is an unexpected result, as the negotiation algorithm was designed to explicitly maximize mutual information in the symbol usages. If negotiation works at all, one would expect it to work in all cases.

A clue to why this occurs can be seen by looking at the standard error bars; for the ten symbol case there is an order of magnitude more standard error in the distribution of negotiation scores than for the five symbol case (0.10 vs. 0.02 bits). In fact, for the ten symbol case four of the 30 negotiation trials produce solutions of more than 3.0 bits of mutual information (maximum possible is $H(C_a; C_b) = 3.3$ bits). On the other hand, there are also three trials of with less than 1.5 bits of mutual information. In comparison, the mutual information values produced by the imitation algorithm all fall between 2.0 and 3.0 bits. So, in fact, negotiation can actually provide a more informative lexicon even in the ten symbol case, but imitation provides a higher *average* score.

This variance can be attributed to the simple fact that the negotiation algorithm cannot backtrack and reassign a new symbol after the agents agree how to label a meaning. Once a particular meaning is assigned to an under-used symbol it may, in fact, cause that symbol to become heavily over-used. If there are many fewer symbols than meanings, there are more chances to rectify any mistakes made in smoothing out the relative occurrence rates of symbols. So, essentially, negotiation degrades less quickly with decreasing lexicon size than imitation only because there are an increased number of chances to fix mistakes made in symbol assignment.

The clear implication is that the negotiation algorithm proposed is not going to be good enough. Imitation will always produce better symbol mutual information on average as long as it's given enough symbols in the lexicon. To get better performance in tasks with difficult perception, like the landmark labeling task presented in section 3.3, it will be necessary to design an algorithm that can either (1) take into account the relative frequency of perceptions as well as that of symbol, or (2) can re-assign symbols even for previously agreed upon meanings.

5. Conclusions and future work

This paper shows that imitation alone is insufficient to develop lexicons useful for a task. Experiments with a simple imitation learning algorithm show that symbol usages will have less average information than is present in the agents' perceptions. The same simulations showed that when perceptual noise is present this effect becomes much worse. An example of lexicon learning on a robotic task presented an extreme case of information loss (only 3 symbols, 1.4 bits mutual information, out of 20 meanings) that motivated the development of a better method.

This paper argues that lexicon learning algorithms should *explicitly* maximize the mutual information be-

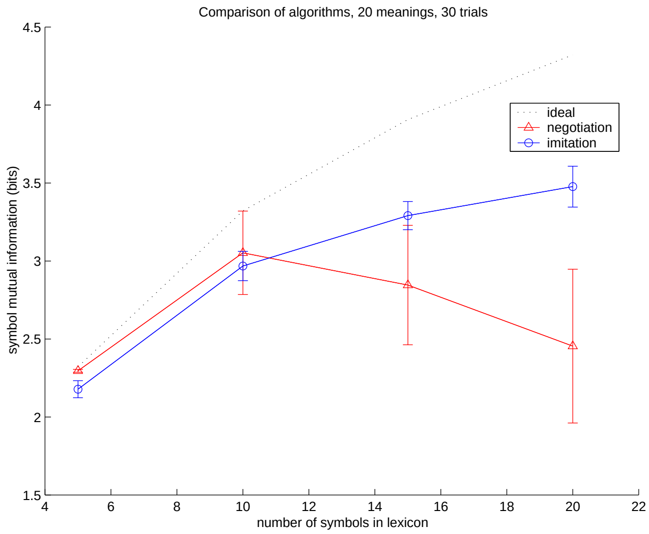

Figure 1: Figure showing how $H(S_a; S_b)$ changes as a function of lexicon size for both imitation and negotiation algorithms. Error bars show three standard errors around the mean value. The ideal line is the mutual information of an equiprobable symbol distribution of the given lexicon size. Plainly negotiation cannot offer better mean mutual information than imitation.

tween symbol usages to prevent this from occurring. Although many lexicon learning algorithms in the literature effectively do this, they do so through hidden or implicate mechanisms, like the example literature analyzed in section 2.3. Because these mechanisms are not explicit they can be difficult to control or have unforeseen consequences in different lexicon learning tasks.

A first step was made toward a useful, information-maximizing algorithm with the proposal of the negotiation algorithm. Unfortunately, the algorithm does not produce better mutual information levels than imitation except in cases where the number of symbols is much less than the number of meanings. This occurs because the algorithm is heuristically approximating the maximization of symbol usage mutual information. Once the algorithm has made a mistake due to that approximation, it cannot re-negotiate a new symbol assignment for that meaning.

The way ahead is therefore clear. A new algorithm should either make a better approximation to mutual information maximization, or it should be able to re-negotiate meaning-symbol assignments that are already agreed upon. The difficulty with the later approach is to be able to guarantee that a lexicon will ever become stable. Therefore, we believe that the former is the best avenue for future investigation. For instance, a new way to calculate an agent's willingness to accept a proposal that uses an estimate of the occurrence rate of the meanings rather than just the occurrence rate of the symbols has the potential to be useful.

Regardless of the method used to achieve it, lexicon learning algorithms would benefit from making information maximization an explicit aim of learning. This approach will hopefully enable the development of minimum performance guarantees and systems which produce higher symbol usage mutual information. In the long term, such research might also shed some light on mechanisms humans and other animals employ to ensure that their symbol usages are informative.

Acknowledgments

The authors thank Christian Freksa at the University of Bremen, Germany for opportunities he provided and Wolfram Burgard at the University of Freiburg, Germany for many helpful discussions about the problems in this paper.

Part of this work was supported by a Canadian & U.S. Citizens' Scholarship and an ATLAS grant, both from the Department of Computer Science at the University of Manchester. Part of this work was supported by a stipend from the International Quality Network on Spatial Cognition, which is supported by the Deutscher Akademischer Austausch Dienst.

References

Balch, T. and Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1:27–52.

Batali, J. (1998). Computational simulations of the emergence of grammar. In *Approaches to the Evolution of Language*, pages 405–426. Cambridge Univerisity Press.

Cover, T. and Thomas, J. (1991). *Information Theory*. John Whiley, New York.

Fleischer, J. (2004). *Route Communication Between Mobile Robots Using Adaptive Landmark Symbols*. PhD thesis, University of Manchester, England.

Garrod, S. and Doherty, G. (1994). Conversation, coordination, and convention: an empirical investigation of how groups establish linguistic conventions. *Cognition*, 53(3):181–215.

Kohonen, T. (1993). *Self-Organization and Associative Memory, 3rd ed.* Springer, Berlin.

MacLennan, B. (1992). Synthetic ethology: An approach to the study of communication. In Langton, C. G., Taylor, C., and Farmer, J. D., (Eds.), *Artificial Life II*, pages 631–658. Addison-Wesley, Redwood City, CA.

Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379 – 423, 623 – 656.

Steels, L. (1996). Emergent adaptive lexicons. In *Animals to Animats 4*, pages 562 – 567. MIT Press.

Steels, L. (1998). Synthesizing the origins of language and meaning using coevolution, self-organization, and level formation. In *Approaches to the Evolution of Language*, pages 384–404. Cambridge Univerisity Press.

Vogt, P. (2000). *Lexicon Grounding on Mobile Robots*. PhD thesis, Vrije Universiteit Brussel.

Werner, G. and Dyer, M. (1992). Evolution and communication in artificial organisms. In Langton, C. G., Taylor, C., and Farmer, J. D., (Eds.), *Artificial Life II*, pages 659–687. Addison-Wesley, Redwood City, CA.