Expressing and Understanding Desires in Language Games

Michael Klein[1,2], Hans Kamp[2], Guenther Palm[3], and Kenji Doya[1,4]
[1]ATR Computational Neuroscience Laboratories
[2]Institute for Natural Language Processing, Stuttgart University
[3]Department of Neural Information Processing, Ulm University
[4]Crest, Japan Science and Technology Agency

Abstract

What did I show? What do we know now, that we did not know before?

Multi-agent reininforcement learning language game

optimization

learn when to speak, whom to address and what to say and when to remain silent or perform a non-verbal action. Further we used to rule-based agents to train a language learner, who observed the effects of the actions of his fellow players and after understanding these effects, would use language to express his own desires. He also uses the trained function to understand the expressed desires of other agents.

Introduction

Verbal communication has a purpose. Language, the means of verbal communication, is used to fulfill a variety of desires - from the most simple and straight-forward (attention, food) to more complex (socializing, find somebody to love, get a good job). This is possible, because utterances have an effect[1], that goes beyond the transmission of its literal content (Wittgenstein, 1953; Austin, 1961). This is especially important when it comes to language acquisition, because children can experience these effects, and thereby learn to *employ* utterances to fulfill their own desires. This observation gives them the motivation to speak and to learn. The more precisely a child is able to express its desires, e.g. to relates the utterances to the relevant properties of the world, the more *effective* its utterances will be.

To increase the effectivity of language (with respect to the goal of bringing about desired states of the world) can be regard as an optimization process ...

...which explains, why children can learn language without a teacher. While many parts of the problem of learning to produce and understand meaningful language, can be learned unsupervised (such as associating syllable sequences to concepts (Klein and Billard, 2001) or forming concept in the cerebral cortex (Klein and Kamp, 2002))...

so far this kind of goal-oriented optimization has not been taken into account.

In this paper we present an approach of how this can be achieved. We include a prelinguistic level of *desires* (embedded in a desire-hierarchy).

Explain more ...

As speaking is not always the best way for a human to fulfill those desires (sometimes other actions are more appropriate, or sometime desires cannot be fulfilled) our approach takes this into account and includes non-linguistic actions as well. The key idea of our work presented in this abstract is, that the agents learns the language by observing other agents using it. Through observation they learn the effects, which certain expressions have in particular contexts. After learning the (context dependent) effects of expressions, the agents can use them to fulfill their own desires (as far as this is possible). With the same cognitive function they understand what another agent wants to achieve by his expression.

In this study we build a computational model of language acquisition, in which two agents with a rule-based language module train one language learner. Using supervised learing, the learner trains a *forward model* which predicts the (context dependend) effects of utterances. To train this model the learner observes the communication of the other agents. He can then use this model (i) to find the right utterance to express his own desires, and also (ii) to understand other agents by mapping speakers' utterances on the state they desire.

To test whether the proposed mechanisms can accomplish these goals, we designed a simulated game environment. We used a game environment, because it allows the agents to form their own desires. Having their own desires, they can learn to use language to achieve their goals. An agent learns a value function that can assign a value to every states telling the agent how desirable it is. Along with a rule-based forward model this function is used to select actions.

Theory

To make an agent express desires, we need to have desires in the first place. This raises the question of how to represent

[1]These effects are sometimes very direct and obvious and sometimes very indirect and subtle

desires. The solution we chose, was to represent desires as *valued* states of the world. For every state of the world, the agent has a positive or negative numerical value expressing how much it desires this state. A function mapping every state on such a value can be called a *value* - function.

Such a value-function estimates how good it is for an agent to be in a given state. The notion of how good is defined in terms of future rewards that can be expected, or, to be precise in terms of the *expected return* (Sutton and Bartho, 1998). The expected return is the sum of discounted rewards, which can be expected in and after a certain state (equation 1) The γ-parameter is the discount factor, which determines the value of future rewards. Given this definition of the expected return, the value function is defined in equation 2.

$$R_t = \sum_{k=0}^{\inf} \gamma^k r_{t+k+1} \tag{1}$$

$$V^{\pi}(s) = E_{\pi}\{\sum_{k=0}^{\inf} \gamma^k r_{t+k+1}\} \tag{2}$$

Reinforcement learning is a suitable method of generating such a function during interaction with the environment.

These methods allow us to determine the *desired state* of every agent in every state: It is the state with the highest value. But not every state can be reached from a particular other state. Therefore, we need a *desire hierarchy* with more desired and less desired states. It is obvious, that the value function gives us exactly what we need.

The theory of meaning we are generating our hypotheses from, can be characterized by the following five central points.

(i) An agent experiences utterances used by others speakers to have effects on the observeable world. These effects of utterances are dependent on the context.

(ii) These effects on the observable world are indirect and are achieved by a more direct effect on unobservable states, such as the mental states of the addressees.

(iii) Linguistics events, i.e. experienced context-depended effects of utterances are used to train a function mapping context configurations and utterances to effects.

Such a function, which predicts a sensation $x(n)$ based on the state $x(n-1)$ (context) and the action $u(n-1)$ (utterance) has been be called a *forward model* (Jordan and Rummelhart, 1992) or *predictor* (Wolpert et al., 2003).

$$x(n) = X(x(n-1), u(n-1)) \tag{3}$$

(iv) A speaker uses a certain expression, because he desires the effects he expect the expression to produce in the present context (according to his experience). He choses the action which will lead to the state of the world, which he desires most.

This output function (or utterance function) maps the observed state and the desired observation into an utterance

$$u(n-1) = U(d(n-1), x(n-1)) \tag{4}$$

This function has been called an *inverse* model (Jordan and Rummelhart, 1992) or a *controller* (Wolpert et al., 2003).

(v) An addressee understands an expression of language, because he has represented the same relation of expression, contexts, and effects. The utterance in the context triggers the representation of an effect, which is likely to be the desire of the agent.

An agent understands an utterance, by understanding the intention of the agent. By using a *predictor*, a function from context $x(n-1)$ and utterance $u(n-1)$ to effects $x(n)$, the agent can know what the other agent is trying to achieve.

The Game

We test our ideas about language acquisition and communication in a multi-agent simulation. In this simulation, *food* grows in certain intervals in *trees*. In the present work we use three trees growing three types of food. Every tree can hold maximally 5 pieces of food, and 3 pieces of food grow simultaniously, once the amount of food in the game is below a certain threshold.

There are three agents in the game. Every agent can store 5 pieces of each food type. Always after a certain time interval one piece of food gets *digested*, i.e. it simply disapears. This is to garantee, that the agents need to act and cannot rest, after they have gained a sufficient amount of food items. However, they do not *starve* if they have no food for a number of time steps, but they get a low reward.

Agents can perform one of the following actions:

- harvest tree (take down all the food)
- give one piece of food to another agent
- ask another agent for a type of food
- do nothing (important!)

Generally, the agents take turns. However, when an agent asks another agents for a type of food, the normal order pauses for one time step, as then it is the turn of the addressee to give (or not to give) the desired object to the speaker.

The goal of the agents in the games is to have one piece of each food types at every time step. Therefore, the reward

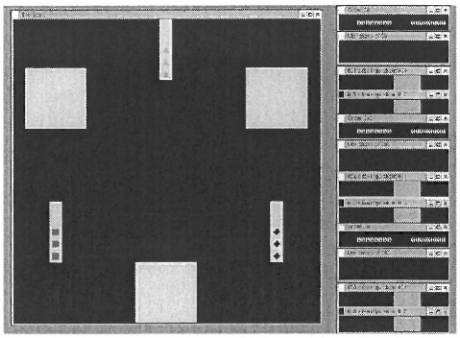

Figure 1: This shows the intitial game state. The long yellow standing rectangles are the trees, each holding 3 pieces of food. The grey squares are the agents. They have the capacity of storing 5 pieces of each food type. The bar on the right displays scores and utterances. The green bars show, which agents cooperated with which other agents in their last move.

function was designed in the following way: Each agents gets a reward at every time step. If an agent has at least one item of every food type, it gets a reward of $+3$, otherwise it gets -1 for every food type which is missing completly in his store at the time step.

The agents in the game are simulated independently. Every agent observes the relevant features of the environment at every time step. Further, every agent has its own memory devices: a short term memory memorizes the complete observable game state (including all utterances) for a constant number m of time steps. The agents interact with the world only by their perception and actions and with each other by perception, actions and utterances.

An utterance of an agent is defined by its content (i.e. which word is used), its speaker (the agent) and the addressee. Who the agent talks to and what he it says is up to him. The possible content is defined by the *vocabulary* of the agents in the game. In the present study it consists of the three words *triangle*, *square*, and *diamond*. The content of an utterance can only be one word. An agent can use only one utterance at every time step.

With respect to their linguistic capabilites, agnts can either be a *teacher* or a *learner*. Teacher-agent do not teach language, but they used a rule-based dialogue system to produce and understand utterances.

To chose their actions (or utterances), the agents predicts the outcome of the action in the present context with a forward-model. The forward model is rule-based for *no action*, harvesting trees, donating objects, and for the verbal actions of the teacher-agents.

These outcomes are evaluated with the value function and the action (verbal or non-verbal) which will bring about the state with the highest value is chosen.

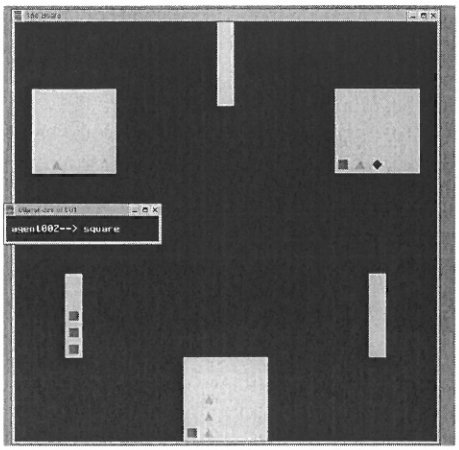

Figure 2: This shows an arbitrary state during the early stages of training. The last action of agent 1 was to ask agent 2 for the square. Obviously this is not the best move. A better move would be to harvest the *square - tree*, as with this action, the agent would get 3 square instead of one.

If a verbal action is selected, and the addressed agent is a teacher, then the addressed agent will give the desired object to the speaker. If the addressed agent is a language learner, this agents applies its *forward model* to the utterance and the game state. With this model he can estimate what kind of change the speaker desires, i.e. he computes the intention from the utterances and the context. In other words, the language learner understand the utterance, because it *wonders* what effect, according to its own experience, such an utterance has in the present context. Using the present state of the game and the estimation of the desired state of the speaker, the addressee then uses a rule-based algorithm to computed which action would bring this desired state of the game about.

Learning Algorithms

The value function maps states of the game to real numbers. A state is given by three 5-dimensional binary vectors (one for each tree) and three 3 x 5 binary matrices (one for agent).

The value function is implemented as a neural network with one neuron for every binary value of the vectors and the matrices. The output of the network is the linear combination of the weighted binary inputs. To train the value function use TD(0) reinforcement learning (Sutton, 1988) as described in equation 5. The term given in 6 is the so-called TD-error, giving distance and direction to the correct prediction and determing the weight changes.

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (5)$$

$$r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \quad (6)$$

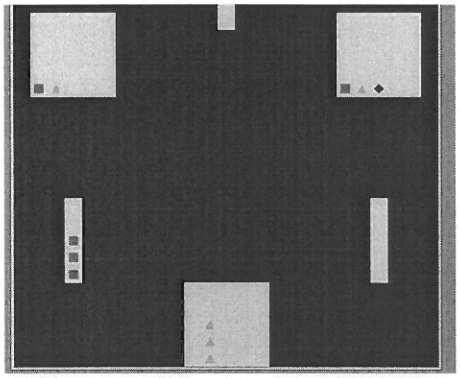

Figure 3: This shows the game state imidiatly after agent 2's reaction to the request of agent 1 in figure 2. We can see, that the square has changed its position from the store of agent 1 to the store of agent 2.

Due to the huge number of possible states, we used a neural network function approximation of the value function. As exploration mechanism we used a *softmax* - method.

In linguistics capabilites of the language learner in the game are represented by a forward model. This model learns the context-dependend consequences of utterances. The context of each utterance is the full game-state, as described above. The forward-model is implemented by a single-layer perceptron, mapping utterancs and game states into game states. We used supervised learning to train this forward model (equations 7, 8, and 9).

$$e_k = y_k^* - y_k \tag{7}$$

$$\delta w_{ik} = \alpha e_k x_i \tag{8}$$

$$w_{ik} \leftarrow w_{ik} + \delta w_{ik} \tag{9}$$

Note that the same forward model can be trained by observing the effects of other agent's utterance as well as the effect of the agents own utterances (i.e. in our approach these two types of predictions are not distinguished, which of course is a considerable simplification).

Results

Concerning the training of the value function, where agents learned, which states are desirable, and whether in certain situation it is better to harvest, donate, speak, or simply do nothing, agents performed extremly well (approximatly at the level of a human player or even better).

At the very early stages of the training, agents selected *no-action* or nonsensical actions very often (such as donating objects to other players without being asked). Sensical, but suboptimal action, as described in figure 2 and 3, did occur in the intermediate stage of training. After training, no more suboptimal action could be detected. Agents used language

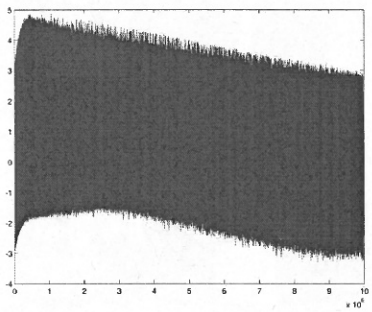

Figure 4: This is the development of the TD-error over 5 * 10^7 time steps.

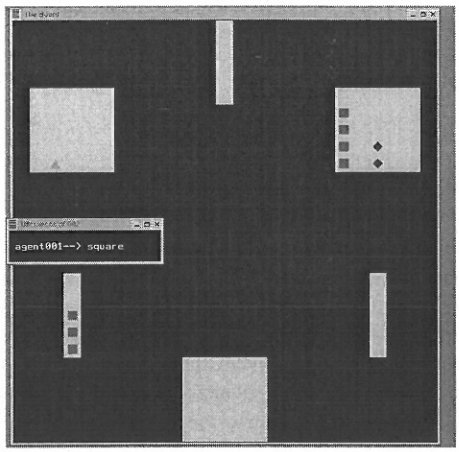

Figure 5: This shows an episode during the language learning. The language learner (agent 2) asks agent 1 for the square although agent 1 does not have one. This is an example of the language learner not able to understand the normal *effects* this kind of utterance.

if appropiate, harvest trees whenever possible and optimal, stoped chosing *no action*, as usually some action or request would improve the state of agent. The value function, as far as observed, gave the appropriate desire-hierarchy. Although the TD-error decrease very slowly, as can be seen in figure 4, optimal performance could already be observed after about a million time steps.

The prediction error decreased very fast to a level close to 100 % with declining occational in the simulation, where only the utterance effects of the two other speakers were used (see figure 6).

Discussion

In this study, we were able to show (i) that general framework of introduced works for the relation between intentions

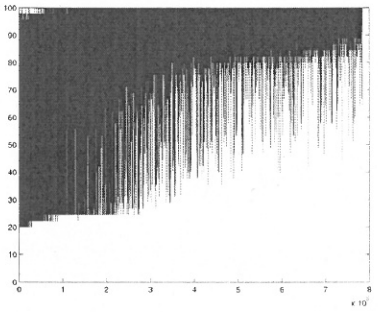

Figure 6: The prediction error of language learning changes over time. This graph showns its development through 5 * 10^6 learning episodes

and utterances can be learned by such a model and that, in a simulated environment, it can be used to communicate well with respect to a task.

The speed of learning
object permance
no causal connection
lacking of generalzing feautures
no *higher* level of representation.

Although in this study, we used the words *triangle*, *square*, and *diamond*, the reader should not be mistaken, that the meanings these words have in the game are not the meanings they have in the real world.

Cannot learn that certain words are used to refer to certain objects as in the work of Luc Steels (Steels, 1996; Steels, 2001).

This kind of generalization is needed to make learning faster and more efficient

While our present approach is restricted to single-word request, the framework is designed to handle multi-word utterances and different kinds of speech acts, such as questions and answers. In such a multi-utterance, multi-speech act game, the framework could show its full potential.

Acknowledgements

This work was supported by the German Academic Exchange Service (DAAD), the German Research Foundation (DFG), and the Telecommunicatios Advancement Organization of Japan (TAO).

References

Austin, J. L. (1961). *Philosophical Papers*. Oxford University Press.

Jordan, M. and Rummelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354.

Klein, M. and Billard, A. (2001). Words in the cerebral cortex - predicting fmri-data. In *Proceedings of the 8th Joint symposium on neural computation - The brain as a dynamical system, San Diego*.

Klein, M. and Kamp, H. (2002). Individuals and predication - a neurosemantic perspective. In Katz, G., Reinhard, S., and Reuter, P., editors, *Sinn und Bedeutung 6, Proceedings of the sixth meeting of the Gesellschaft fuer Semantik, Osnabrueck, Germany, October 2001*.

Steels, L. (1996). Perceptually grounded meaning creation. In Tokoro, M., editor, *Proceedings of the International Conference on Multi Agent Systems*, pages 338–344. AAAI Press.

Steels, L. (2001). Language games for autonomous robots. *IEEE Intelligent systems*, pages 16–22.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44.

Sutton, R. S. and Bartho, A. G. (1998). *Reinforcement Learning - An Introduction*. MIT Press.

Wittgenstein, L. (1953). *Philosophical Investigations*. Blackwell.

Wolpert, D. M., Doya, K., and Kawato, M. (2003). A unifying computational framework for motor control and social interaction. *Phil. Trans. R. Soc. Lond.*, 358:593–602.