Multimodal Corpus Collection for the Design of User-Programmable Robots

Joerg C. Wolf **Guido Bugmann**
Robotic Intelligence Laboratory
School of Computing Communications and Electronics
University of Plymouth
Drake Circus
Plymouth PL4 8AA, U.K.
{joerg.wolf, guido.bugmann}@plymouth.ac.uk

Abstract

In order to design a robot able to learn from its user, the example of card game instruction is investigated. A detailed description of a setup for the collection of a human-to-human multimodal instruction corpus is given. This setup uses touch screens and will also provide a base for the human-robot instruction interface to be designed. Preliminary results on learning dialogues are given and issues of corpus transcription, annotation and analysis are discussed.

1 Introduction

In the future, intelligent robots may become part of daily life. Robots are already entering our environment as interactive toys. Robots will manage the household or an office environment as autonomous agents. Future service robots can not be completely pre-programmed by the manufacturer. There are far too many possible tasks. In order for these robots to successfully learn and interact with people from the general public, they must be *programmable by anybody* (naive users / without training) and not just by engineers, roboticists and computer scientists. The user does not even have to be IT-literate. The design of a "user programmable" system is the subject of this research.

A user-programmable robot must use an interface that is natural to the user. The best way to design a truly easy-to-use interface is by examining the interaction between people. By observing instructions from human teachers to human students, guidance is sought here for the design of a robot acting as the student.

Previous work carried out by our laboratory on the Instruction Based Learning project (IBL) (Kyriacou, 2004; Bugmann *et. al.* 2004) has shown that it is possible to extract information from a representative sample of the teacher's utterances (the "corpus") in order to:

- Identify primitive procedures that the robot has to be able to carry out (the robot's "prior knowledge")
- Write and tune a speech-recognition software to call and combine these primitive procedures.

This approach to the definition of the robot's functionality and natural-language interface (NLI) has been described as "corpus-based robotics" (Bugmann *et. al.* 2004) and is outlined in figure 1.

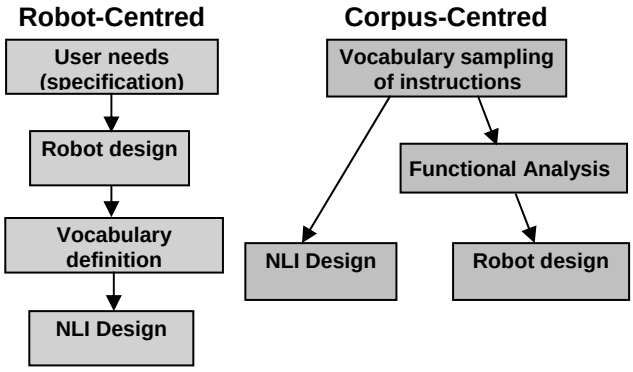

Figure 1: Robot vs. Corpus-Centred Natural Language Interface (NLI) design. In the Corpus-centred approach, the content of samples of instructions between humans defines at the same time the vocabulary to be dealt with by the speech interface and the required functionality of the robot. In the robot-centred approach, the functionality is defined first, then the access vocabulary, then the NLI.

The IBL project focused on route instructions given to robots. A dialogue such as the following was possible between the user and the robot:

User: "Go to the University."
Robot: "How do I go there?"
User: "Take the third turning to the left..."
Robot: "Next instruction please."
User: "...take the third exit off the roundabout..."
Robot: "Next instruction please."
User: "The University will be on our right."
Robot: "OK, it's done."

Since the IBL project was using route instructions, the resulting system was developed to deal with sequential instructions, and could not handle other forms of instructions, such as general rules, which apply at any time during the task, such as "Stop at the petrol station if you run low on petrol". The system could not deal with conditionals, such as the one above, that were not found explicitly in the corpus (Lauria *et al.*, 2002). In route instructions, sentences starting with "if" instructions are generally just a colloquial way of expressing a sequential instruction, as in the following example from the IBL corpus: "...okay if you carry on straight along this road and if you take the third left you will go over a bridge..."

Therefore, to develop a more general instruction system, there is a need for looking at a different application, where instructions not only include sequences, but also other instruction structures. In imperative programs these would be decisions and repetitions. However, in the declarative paradigm, programs consist of lists of goals and a set of rules (see e.g. PROLOG). It is unclear which paradigm is a more useful representation of human instructions. This is one of the questions that need to be addressed by analysing a new corpus of instructions in a different domain (see section 2).

Furthermore, previous research in our group focused purely on verbal instructions which are well suited to communicate rules and sequences of actions, but are less well suited for other aspects of instructions. In practice, many tasks are explained using a mixture of verbal instructions, gestures and demonstrations. Thus, a truly natural interface between human and robots must be multimodal. This is one of the features to include in this project.

In the following section, we describe the selected instruction domain and the particular setup designed to facilitate the capture the human subject's multimodal behaviour and their reproduction by a robot.
In section 3 we describe the corpus collection protocol and in section 4 (very) preliminary results are given[1] along with consideration on its semantic annotation. Section 5 is the conclusion.

[1] *The corpus is currently being transcribed and annotated*

2 Experiment Design

2.1 Instruction domain

The criteria for the selection of a new application/task are as follows. i) The task should contain a wide range of instruction types. ii) The task should be scalable from simple to complex. iii) The vocabulary should be restricted to a domain.

Given these constraints, game instruction seems to be a good choice. In particular, card games come in a great variety of type and complexity, yet their vocabulary is restricted.

We investigated all two player games listed in "the Oxford A-Z of Card Games" (Parlett, 2004), and defined a simple measure of complexity as being the sum of the number of rules stated in Parlett's game descriptions. The more rules a game had the more complex it was assumed to be. We selected the national Italian card game "Scopa", since it has intermediate complexity that is typical for card games and is not commonly known in the U.K. Scopa is a fishing-type card game. A fishing game means that there are several cards face-up on the table, and the players have to match cards in their hand with the cards on the table. Matching cards on the table can be captured by the player in order to score.

2.2 Multimodal interface

In future robots, multimodal interfaces will require complex sensory processing, such as gesture and face recognition. As this project focuses on the problem of learning, we decided to devise a simplified interface that would still allow natural communication with human users.

Our solution to the problem is to use a touch screen that allows at the same time to acquire human gesture information by the robot (without complex sensory processing) and execution of game moves (without complex actuators). The screen represents the world as the robot would see it through it's vision system. The user is able to point at and manipulate objects on the screen as a demonstration of how to do the task. At the same time the user gives verbal instructions. Touch-screens have been used in multimodal human-robot interfaces for different applications, for example by (Perzanowski *et al.*, 2001), or for investigations in human communication (De Ruiter *et al.*, 2003)

A great advantage of using a screen representing the robot's world is that the robot can be simulated, while the interaction and interface to the robot does not change. It also allows focusing research on human-robot interfaces without having to build a robot.

The software used to display the playing cards is based on the Qt (Trolltech®[2] 2005) and the OpenGL®

[2] *Qt is a trademark of Trolltech in Norway and other countries. http://www.trolltech.com/*

API[3] and is platform independent. Qt is a cross-platform C++ GUI development library. OpenGL® is a standard for a 3D/2D cross-platform Graphics API. The playing cards are described as objects with parameters such as size, texture, position, orientation, static or movable. Therefore the system can be used not only for card games. The display software could display any real world object that the robot knows about. The user can manipulate these objects intuitively. The computers used for displaying the cards are linked to a server via TCP/IP. All events of object manipulations are logged at the server and forwarded to all other connected clients. So if objects are moved on one screen, they move on the other screens as well.

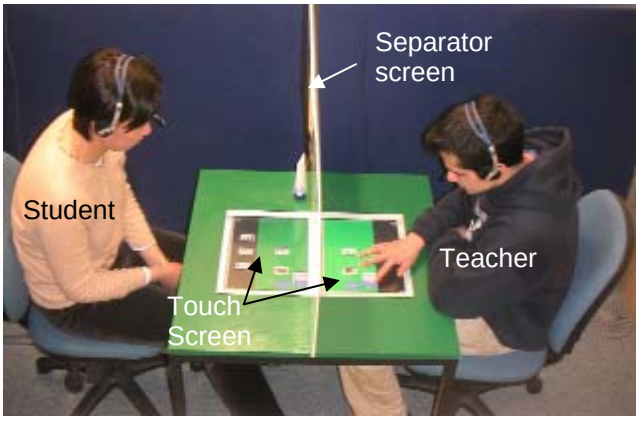

Figure 2: Setup for corpus collection

2.3 Corpus collection

A teacher and a student sit at a desk (Figure 2). The two are separated by a screen so they can not see each other. The desk has touch screens build into its surface. Playing cards are shown on the screens. The cards can be moved around on the screen by touching and dragging them around. Both players have a common area for cards, and an area that can only be seen by one player (black area on the touch screen in figure 2) symbolizing the cards in the hand. The common area is located near the screen and represents the virtual playing-table. The teacher will explain a card game to the student. The interaction is filmed and the dialogue recorded. To ensure high quality recordings, the subjects wear headset microphones. The coordinates and movements of the cards on the touch screens are recorded simultaneously. The data can be synchronized with a time stamp. This simultaneous recordings of voice and touch screen data from object manipulations constitutes a multi-modal corpus.

[3] *OpenGL® and the oval logo are trademarks or registered trademarks of Silicon Graphics, Inc. in the United States and/or other countries worldwide*.

We recorded 21 dialogues between teachers and students. Students who learned the game in one session became the teachers in the next. In the design of the protocol we tried to avoid two forms of bias, the vocabulary bias and the instruction strategy bias.

Pilot studies revealed that a teacher subject tends to use expressions and methods similar to those used when he/she was taught. To avoid this bias, we decided that the initial teacher (Student S0 at the top of the tree) would learn the rules of the game from a set of rules written on separate sheets and presented in random order. Subjects usually proceed by re-ordering the sheets to help learning the game. Two sets were used with different words for the same rules.

In order to maintain the chain if a subject decided to drop out, the experiment was designed in a tree structure where one teacher teaches two students, and then these students become teachers themselves. Figure 3 shows one of two trees used in this experiment. Example: Teacher S0 teaches student S1 and S8. After that S1 becomes a teacher and teaches S2 and S3.

We left at least one day between learning the game and having to teach it. This generally led to a fading of the memory of the precise words and order of instructions. Thus the chain design is expected to reduce the bias in vocabulary and lead to an increased variety of instruction styles in the corpus.

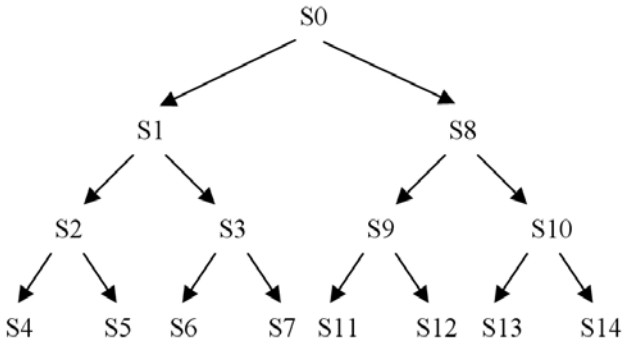

Figure 3: Tree of teaching dialogues. Two trees of this type were used to record dialogues. There are 14 dialogues in each tree, represented by the arrows and organized in three layers. S*i* is the subject number *i*.

3 Preliminary results

As a general observation, the use of a separation screen between teacher and student had the expected effect, in that gesture communication was very much restricted to the touch screen. Very few gestures "in the air" were observed.

The next observation is that the length of the dialogue for explaining the same card game decreased along the chain of dialogs (see figure 4). It appears that the way the game was explained became more efficient in lower layers of the tree.

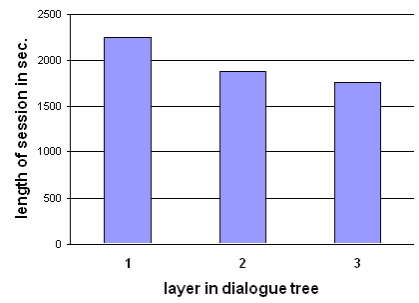

Figure 4. Dialogue length vs. layer

A typical conversation extracted from the corpus (ts_session19 03:56).

…
Teacher: "There is an orderly deck of cards…All the numbers are their usual value. Other than the fact than Jack is 8, Queen is 9 and King is 10…and ace is 1."
Student: "right"
Teacher: "So you need to remember that, obviously for when you are pairing or capturing cards."
Student: "So Jack was 8 you said."
Teacher: "Yeah"
Student: "And ehm, Queen was 9."
Teacher: "Yeah"
Student: "And King is 10."
Teacher: "Yeah. And Ace is low, number 1."
…

A first look at some transcripts suggests the presence of at least two types of primitive functions in the instructions:

- Knowledge management functions ("A king is worth 10")
- Action functions ("Put down this card")

These kind of functions will need to be implemented in the learning robot. Finally, we also noted the occurrence of:

- contradicting statements
- underspecified statements
- mixed up order of instructions

These features are likely to represent challenges for the design of dialogue and knowledge management components of the robot-student.

4 Discussion

4.1 Multimodal transcriptions

Speech transcription can done using tools such as Transcriber[3]. This produces a time stamped XML text corresponding to a recorded sound file. We are currently investigating if there are similar tools for the transcription and annotation of signs or gestures done in a card game on a touch screen. Otherwise, dedicated software will have to be developed, possibly inspired by (Bird and Liberman, 1998). The task of such software is to annotate the raw recording of trajectories on the screen with high-level "sign tags", such as *pointat(AceClubs) or turnover(AceHearts).*

The multimodal recordings and transcriptions in these experiments are linked with time-stamps. This requires the recordings to be started simultaneously. Similarly (Knut Kvale *et al.* 2004) uses timestamps to synchronize inputs from touch screen and voice. Transcriptions are commonly done in XML. However, there is no widely accepted standard for multimodal transcriptions. We are in the process of reviewing transcription and annotation tools.

4.2 Semantic Annotation

The purpose of annotation of this multimodal data is to provide reference data for testing the system to be developed and identifying the semantics in a formal format. Two streams of data are coming into the robot: utterances and touch screen inputs.

The touch screen inputs are in the form of trajectories of card movements and must be converted into a "symbolic" format, such as *moveto(KingHearts,table)*, referred to as signs. Both, voice and signs have timestamp and duration.

Synchronizing the two allows resolving deictic references. Figure 5 shows an example, where the teacher says "this one" and starts pointing at a card. The gesture *F* means that the card was touched by the user. The gesture *M* stand for subsequent movements (wiggling) of the card, which would be recognized by the student.

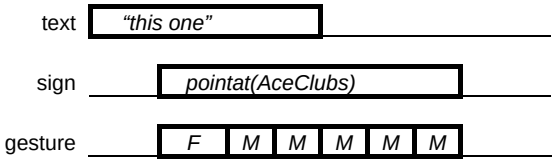

Figure 5: Timing diagram of multimodal inputs

One point worth considering is that the annotation scheme is tightly coupled with the system's concept of operation. For instance, one could decide that any utterance by the teacher requires an action by the robot. Based on the primitives noted in section 3, these actions would then be knowledge manipulation actions or actual actions on the cards in the game.

[3] *http://www.etca.fr/CTA/gip/Projets/Transcriber/*

5 Conclusions

In order to design a robot able to learn from its user, the example of game instruction is investigated. A detailed description on a setup for human-to-human multimodal instruction corpus collection is given. The use of touch screens as the "game table" simplifies the recording of gestures and will greatly simplify the design of the learning robot, which will essentially be a software agent.

The transcription and analysis of the corpus has been discussed and is likely to require new tools, especially in the area of signs definition and transcription. The semantic annotation scheme needs to be defined carefully, as it links to system design. In addition, it is possible that the designed agent will have to reproduce some of the features of the human thought processes engaged during learning and playing, to give the robot the ability to learn from natural instructions. Information on these issues is expected to be produced by the analysis of our corpus of human teacher-student dialogues.

References

Bird, S. and Liberman, M., (1998), Towards a formal framework for linguistic annotations. *Presented at the ICSLP, Sydney*

Bugmann,G., Klein, E., Lauria, S., Bos, J. and Kyriacou T., (2004) "Corpus-Based Robotics: A Route Instruction Example" *in Proceedings of IAS-8*, 10-13 March 2004, Amsterdam, pp. 96-103.

De Ruiter, J. ,P. , Rossignol, S., Vuurpijl, L., Cunningham D.W. and Levelt, W.J.M., (2003). SLOT:A research platform for investigating multimodal communication. *In Proc. Of Behavior Research Methods, Instruments & Computers 2003*, 35(3),408-419

Miura,J. , Yano,Y. , Iwase,K. and Shirai, Y., (2004), Task Model-Based Interactive Teaching, *In Proc .of IROS 2004 Workshop on Issues and Approaches to Task Level Control*, Sep. 28, Sendai, Japan, 2004.

Kvale,K., Knudsen, H., E. and Rugelbak, J., (2004) "A Multimodal Corpus Collection System for Mobile Applications", *LREC2004 Workshop, Lisboa, Portugal*, 24.5 - 30.5.2004.

Kyriacou T., (2004), Vision-Based Urban Navigation Procedures for verbally instructed robots. PhD Thesis, University of Plymouth, U.K.

Lauria, S., Kyriacou, T., Bugmann, G., Bos, J. and Klein, E. (2002). Converting Natural Language Route Instructions into Robot Executable Procedures. In *Proc. of the 2002 IEEE International Workshop on Robot and Human Interactive Communication* (Roman'02), Berlin, Germany, pp. 223-228.

Leech, J. and Brown, P.,(2004), The OpenGL® Graphics System: A Specification, Version 2, Silicon Graphics Inc.

Parlett, D., (2004), "the Oxford A-Z of Card Games", Oxford University Press, Second Ed.

Perzanowski, D., Schultz, A., C., Adams W., Marsh, E. and Bugajska, M., (2001), "Building a Multimodal Human-Robot Interface", *IEEE Intelligent Systems, 16 (1), IEEE Computer Society*, 16-21.