Instruction-Based Learning for Mobile Robots (IBL)

1. Background

The IBL project was a collaboration between the School of Computing at the University of Plymouth and the Institute for Communicating and Collaborative Systems at the University of Edinburgh (Dr. Ewan Klein, separately funded under GR/M90160).

The aim of the project was to explore the possibility for naïve human users of an assistant robot to instruct the robot on how to perform a given task using unconstrained natural language (NL). This corresponds to the creation of a machine language program from verbal instructions.

The need for such functionality arises from several considerations: i) Assistant robots are multifunctional machines, where the details of the required tasks are only known to the user. It is thus not possible to fully pre-program such robot in factory. ii) A wide range of advanced technical skills are required to program a robot. Users are generally robot-naïve and it is not economically viable to dispatch a technician to the user's home each time a program modification is required. iii) Robots that cannot adapt to the needs of their users have no commercial future.

Comparatively little research has been devoted to Instruction-Based Learning (IBL) (Huffman and Laird, 1995; Crangle and Suppes, 1994; Torrance, 1994; Matsui et al.,1999). All these previous approaches used a constrained language that the user had to learn. Thus, a specificity of the Plymouth-Edinburgh IBL project is the handling of utterances that are natural to the user. This imposed wide-ranging constraints on system design (see next section) and led us to propose the term "corpus-based robotics" to describe our approach.

At the start of the project, we decided that we would neither attempt to solve specific problems in natural language processing or in robotics, but attempt to assess if the current state of knowledge enabled the design of IBL systems, and if not, identify the bottlenecks. To that effect, we decided to build a working system in the restricted domain of route instructions with the best available tools. Following objectives were defined.

1. To develop principles and an architecture for instruction-based learning in the context of natural language route instructions to a mobile robot.
2. To build a dialogue manager that mediates between the user input and the robot's procedure building capacities.
3. To develop robot sensory and motor functionality to the level required for interpreting route instructions.
4. To determine the usability of a commercial speech understanding interface in task-related dialogues between users and an intelligent embodied agent.

These objectives were met. A working IBL system was built that was used in public demonstrations and was tested at the end of the project with naïve subjects. More details on the work completed and the lessons learnt are given in the next section. Note that specialized developments in natural language processing components of the system are covered in the report on GR/M90160 by the Edinburgh team.

2. Work undertaken and main results

2.1 Experimental environment and robot

The practical aim of the project was to enable subjects to instruct a robot that would then navigate to its destination by following the instructions. For that purpose, we constructed a miniature town of size 170 cm x 120 cm town, including road elements such as intersections, a roundabout, a bridge, etc. and buildings recognizable from attached logos and names, such as Tesco, Hospital, Post-Office, etc.

A miniature robot was purposefully built for that project robot, based on a 8 cm x 8 cm robot football robot on which a colour video camera and a 2.4Ghz video transmitter were mounted. The control system of the robot was redesigned around a triple PID circuit, as very low speeds of less than 10cm/sec are required in that application. The design of this robot is now shown as an example of application on the website of the robotics company which provided the robot base (www.merlinsystemscorp.co.uk).

The robot operated in a "remote brained" architecture, with image processing taking place on a PC linked wirelessly to the robot. Images of the robot following a route can be seen on the project's web page (IBL-Web). Towards the end of the project it became more difficult to obtain interference-free images, due to the increasing use of the 2.4Ghz band by wireless computer networks. In the future, the use of a larger robot with on-board image processing may be preferable.

2.2 Corpus collection and analysis

For this project it was decided to tune a commercial speech recognition system (NUANCE, www.nuance.com) to the domain of route instructions in a way that would, on the one hand, enable users to speak without apparent constraints, while, on the other hand, maximizing speech recognition performance through a restriction of the lexicon and the grammar. A corpus collection was undertaken to determine the content of natural language in this domain. The corpus contained 144 routes produced by 24 paid subjects instructing 6 routes each. The protocol was designed to induce subjects to speak as if speaking to another human (Bugmann et al., 2001). The sessions were filmed and the instructions were recorded digitally with a good quality headset microphone. The corpus of recorded instructions is available on the web, with corresponding transcripts and functional annotations (IBL-Web).

Lexicon. The corpus contained of 6600 words of which approximately 330 were distinct words. However, the lexicon was not closed. Data analysis showed that one new word was expected to appear for every two route instructions (Bugmann et al., 2001). This is a general problem with corpora in a limited application domain. In the navigation domain, such "out of grammar (OOG)" expressions are unavoidable, consisting for instance of personal names or landmark names. Thus, it is vital for a real-world application that the robot has the ability to learn new words. There is no accepted method yet to achieve this, but the OOG problematic was not investigated further.

Functional analysis. One of the tasks in the project was to convert natural language instructions into action descriptions executable by the robot. To support this conversion, robot-executable primitives must closely correspond to primitive actions found in NL descriptions.

There was not published method for extracting primitives from NL instruction and a method had to be defined for this as well as for annotating transcripts (Kyriacou, 2004, Thesis). We found 15 primitives in the corpus such as "go, location_is, destination_is, go_until, etc" (Bugmann, 2004). Such functions would not necessarily have been defined as "primitives" by a roboticist, as they are complex functions requiring artificial vision and planning, and they do not constitute a minimal set of functions for robot navigation. They are conditioned solely by the way humans describe tasks to humans in the conditions were the corpus is collected.

As for OOG words, "OOG primitives" were to be expected. Data analysis showed that a reference to a new function is expected for every 35 instructions (Lauria et al., 2001). This poses a more complex problem because not only the expressions referring to the primitive must be added to the grammar, but also the execution code for the new primitive must be created automatically. In certain cases, learning by example may prove useful (see e.g. Schaal et al., 2003) but this is probably not a general solution. We made no attempt to solve this problem during the project, but it will need to be addressed at some point.

2.3 Primitives programming

Requirements: In human language, references to actions are highly under-specified. For instance, we say "take the next left" without specifying how many meters must be walked and how many degrees to rotate, because the listener is assumed to be able to take the next left without such details. Thus, for "understanding" NL instructions, robots need execution capabilities close to those of normal humans listeners. Primitives must be able to gather missing information from the environment in a way that enables coping robustly with a variety of turns and other actions.

Implementation: In the project, 15 primitives had to be pre-programmed. This is a huge undertaking for a project of this size. Fortunately, it was found that all primitives had the same algorithmic structure, a search-and-act loop, and could be built from a small number of "machine primitives" (Kyriacou et al, 2004 – submitted).

Among the machine primitives, the most critical was the one dealing with visual recognition of landmarks such as intersections or buildings. While much work has been done elsewhere on road lane following, the problem of detecting and taking side roads in urban environments has only recently started to be addressed, and this by few groups (Gregor et al., 2002). In this project, a much larger variety of intersections had to be handled, such as cross-roads, T-junctions, roundabouts entrances and exits, etc. Different methods were tried and the most robust was a new task-guided template matching method (Kyriacou et al., 2002). It involves road-surface segmentation through its features in the chromaticity space, then an inverse-perspective mapping implemented as a fast look-up table mapping, then the matching of task-specific road templates. As an intersection is located out of the field of view of the camera at the most critical moment, when the robot is very close to it, a novel short-range temporary spatial memory was implemented.

The problem of recognizing buildings appears mainly in the final instruction of a route explanation. This is often a statement like "and you will see it there on your left". This instruction is the one requiring the most autonomy from the robot. It is highly under-specified and the robot needs to visually locate the destination and then plan a path towards it. In our miniature town, we have not attempted to identify and locate buildings due to a lack of time. Instead, a coloured strip was placed at the foot of buildings to signal their positions. This enabled testing the robot on complete routes. In a real urban environment the final instruction would pose vision and control challenges that are beyond current technical capabilities.

These recognition methods proved to be very robust and, out of 443 primitive calls in the corpus, 5 (1.1%) failed due to intersections or buildings not being localized correctly.

Knowledge representation and instructions verification: In an IBL system, it is important to verify that the spoken instruction has been truthfully converted in a machine language program and that the created program is executable. While the former can only be verified through inspection by the user, the latter can be verified internally, using knowledge about primitives. In this project, this knowledge is represented as "Precondition-Action-Consequence (PAC)" triplets (Lauria et al., 2002a). The precondition defines elements in the current state of the robot that must be satisfied before the action can start. The consequence is a modification of the initial state by the action. This knowledge is encoded in the form of a "prediction" function associated with each primitive, alongside the "action" function that controls the robot's motion. The prediction function allows simulating the sequence of actions in a route instructions and detecting inconsistencies, occurring for instance when the consequence of one action is incompatible with the precondition of the next action. In practice, the "PAC-test" mainly detected when the initial position of the robot needed specifying. Another role of the prediction function was to verify that the parameters combinations extracted from the user's utterances were among those that the primitive could handle. For instance, the primitive "turn" could not handle the parameter "direction=forward". Such cases resulted in an error message "unknown parameter xyz" or "invalid parameters combination xyz". Such errors were frequently detected (see section 2.5).

2.4 Architecture

The system was built around the freely available Open Agent Architecture (OAA) designed for distributed interactive systems (www.ai.sri.com~oaa). Each component of the systems is an "agent" that communicates with other agents via a "facilitator". In the IBL system (Lauria et al., 2001), agents dealing with natural language processing (NLP) and human-machine dialogue resided on a Sun workstation running the Solaris Operating System (OS). These were collectively called the Dialogue Manager (DM). Agents dealing with the conversion of semantic representations into robot-language programs and with robot control resided on a PC running the LINUX OS. These were called the Robot Manager (RM). Both machines were linked through an Ethernet connection.

NLP components were written in PROLOG and robot component were written in PYTHON and in C++. The interpreted language PYTHON was selected as it enables the creation of a new program as a text file that can be imported as an immediately executable procedure, without the need for a recompilation step. This was particularly well suited for creating new programs on the basis of verbal instructions. The C++ language was used to create computation-intensive low-level vision primitives.

The RM kept track of the progress of the dialogue with the user and, after each utterance, sent to the RM a Discourse Representation Structure (DRS) representing the whole history of the dialogue.

Translating user commands: A significant effort went into the development of a new software (SEMPRI) for extracting calls to robot primitives from the semantic representations of the instruction contained in the DRS. This is a many-to-one mapping problem, with many different references to a primitive being mapped to a single primitive. For instance, the expressions "take the next left, turn left, take the first turn left, etc" must all be mapped to turn(direction="left", ordinal="first"). For that purpose, a new Procedure Specification Language (PSL) was developed in which rules can be defined to associate a given part of a DRS structure to a given function call (Lauria et al., 2002b). It was found that approximately 200 rules were required for the 15 primitives in this application. Such software allows interfacing a robot application with standard NL tools.

Learning: An important role of the RM was to manage the learning procedure. When the DRS contained a command from the user to navigate an unknown route, the RM sent an error message back to the DM which then initiated a learning dialogue with the user: "Can you explain me how to go to XYZ". From that point onwards, all users commands classed as "replies" in the DRS were converted into a new PYTHON program by the RM (provided that all the internal tests were passed). The created program was built with the same two-parts structure as primitives. The first part is the "action" function which is assembled from the action functions of the primitives referred to in the instructions. The second is the "prediction" function which is assembled from the prediction functions of the primitives (Lauria et al., 2002a). The use of this standard structure enables procedures created under the user's supervision to be later re-used when more complex procedures are explained. This is one of the exciting potential of IBL.

Re-using previous instructions: To explore this possibility, the subjects in the corpus collection were instructed to refer to previously instructed routes when this simplified the explanation of new routes. This turned out to pose complex problems.

The first difficulty was that references to previously instructed route are difficult to detect in an utterance. In one third of the cases, subjects referred to previous route implicitly, e.g. via a landmark that was part of a previous route. For instance, when a subject said "go to the roundabout", it was unclear if this referred to a roundabout that was just in front of the robot or a roundabout further away that could be reached using parts of a route previously instructed. In two third of the cases, the destination of a previous route was explicitly mentioned "start as if you were going to the post-office" but in half of these cases, the sentences had structures that prevented correct recognition.

Interestingly, human subjects listening to the instructions detected only 55% of references to previous routes. Only when they started to drive the robot by remote control along the route, did they notice that there was a problem.

The second difficulty is that almost all references to previous routes required a partial use of the learnt instruction sequence: e.g. "take the route to the station, but after the bridge turn left". One of the problems is that the bridge may not even be mentioned in the instruction of the route to the station. No definite solution has been found to that problem. One idea was to implement a multi-threaded concurrent processing scheme where the robot would "follow the road to the station" and at the same time "try to find the left turn after the bridge". The second process would remain the sole active as soon as the turn is found (Lauria et al., 2002b). It remains to be seen if this solution is general enough, but it is interesting to note that the way users express themselves could end up dictating the (multi-threaded) computational architecture of the robot controller.

Dialogue model: A very simple dialogue model was implemented in the IBL system. First, if speech recognition had a score well below a given threshold, the user was asked "Can you repeat please?". This was not necessarily effective if the cause of the problem were the use of unknown words or grammatical structures not covered by the system. After a few unsuccessful attempts, some subjects tended to produce commands in a more telegraphic style leading to sentences definitely not covered by the grammar. Thus, there is a need, in future systems, to design the dialogue system in such a way that the user is "led in the right direction" in terms of grammatical forms.

If the score was only slightly below threshold, the DM asked the user to confirm the interpretation: "Did you mean to say XYZ?" The user was then expected to answer with "yes" or "no". This proved useful to correct speech recognition errors.

Finally, if the RM detected an error, such as an incorrect parameter, the DM generated the message "I did not understand what you said". This was not very informative for subjects and led to some frustrating interactions.

A fundamental floor of such an approach is that clarification dialogues are initiated by the system, based on internal error detection mechanisms. There were cases where wrong interpretations passed all internal tests and were converted in a program without the subject being able to do anything about it, actually not even noticing it.

2.5 Evaluation tests

One half of the instructions in the corpus were selected at random for use during the development of the system and the other half for its evaluation. Evaluations were conducted either with recorded instructions from the corpus or with human subjects interacting with the completed system. Some subjects were assessed for their ability to follow instructions in the corpus. Overall, tests using recorded instructions as input showed poor performances while tests with human users were much more encouraging.

Systems performance in route learning.

- There was a 40% word error rate when the speech recognition system was fed with recorded instructions. This high number came as no surprise, given that the grammar used for speech recognition only covered 60% of the corpus (spoken language contains many "un-grammatical" structures). However, a limited test with the final dialogue system where one of us (GB) spoke instructions from the corpus showed a word error rate as low as 3%, where none of the error was critical for the understanding process. The main reasons for the drastic improvement were that the utterances were spoken clearly, errors in automatic chunking were fixed, and disfluencies (starters, fillers and repetitions) were removed. Further, the dialogue system gave the opportunity to correct utterances. So, it is possible that with a minor increase in the linguistic discipline of users, speech recognition can become quite effective.
- When using recorded instructions as input, 68% of routes were not converted into programs due to the internal detection of errors. The cause for this high error rate lies in 1. Word errors leading to incorrect parameters passed to primitives. 2. Failure to recognize references to previous routes (see section 2.4). 3. Failure to recognize references to the destination in the final instruction. 4. Unintelligible instructions.
- An important finding is that, among the 32% of programs created, only one half were correct in the sense that the robot would have reached its destination. The fact that 16% of erroneous programs were created is indeed unacceptable for a real-world application. This points again to the need for mechanisms enabling the user to probe the knowledge acquired by the robot, in the same way as a teacher would check that the student has correctly interpreted his instructions.
- It should be noted that tests with recorded instructions do not allow clarification dialogues with users, so that any detected error becomes a fatal one. By conducting tests with human subjects a better image of the performance of the system can be obtained. We used 6 paid subjects, where each had to explain 6 of the same routes used in the initial corpus. They received no prior instruction on the use of the system. For the first instruction they were moderately successful, managing to create a program for only 60% of routes. However, they adapted rapidly and, after the 3rd route instruction, achieved a 100% success rate. Thus, the adaptability of users can and should be exploited in the conception of future IBL systems.

Systems performance in route following.

- In order to assess the performance of the primitives independently from the performance of the NLP components, the instructions in the evaluation corpus were converted by hand into PYTHON programs. When the robot followed these programs, it succeeded in reaching the goal in only for 63% of the routes. In 29% of the routes, it failed because there were errors in the instructions or ambiguous statements. Quite frequently, subjects confused right and left, referred to a crossroad as a T-junction, etc. Interestingly, in 3% of the routes, the robot failed because the instruction referred to a parameter combination that was not encountered in the development corpus (“cross the car park”), or referred to a function that was new to the system (“bear left”). This illustrates the problem posed by out-of-grammar expressions in domain-specific IBL systems. In 5% of routes, the robot failed to reach the goal because of a limitation of its vision system, e.g. missing a landmark out of its field of view in a curve.
- In contrast, given the same recorded instructions, human subjects succeeded in 83% of the routes. When they failed, it was either because they had not listened carefully to the instructions or because there were fatal errors in the instructions, e.g. turn “left” instead of “right”. Interestingly, in many cases of erroneous instructions, subjects still reached the goal e.g. by noticing the error while navigating (“why am I entering a dead end now?”) or by stopping to follow instructions as soon as the destination was in sight. This suggests that human-human communication can be quite lax, yet effective, because the listener has the ability to correct errors. In that case, human-robot communication will become truly effective only if the robot also posses such an autonomous error correction capabilities. For safety reasons however, the robot would need to inform the user of its decisions prior to execution.

2.6 Summary of the findings

Overall, this project produced a wealth of data and highlighted many problems that need solving before robust IBL systems can be built. These offered too many avenues of research, and only problems critical for the production of a functional system for final tests were pursued. The main lessons are summarized hereafter.

- To base robot design on the functional analysis of a corpus was an enlightening experience and we attempt now to promote the approach as “Corpus-Based Robotics” (Bugmann, 2004). The great advantage of this approach is that it specifies clearly what problems need to be worked at and when they can be considered solved.
- A fundamental problem raised by this approach is the need for dealing with Out-Of-Grammar situations (words or actions). Any interactive system will have this problem which has no obvious solution.
- References to previously taught procedures are often implicit and involve partial re-use of the procedure. This posed NLP and program architecture problems that need to be explored further.
- Current approaches to clarification dialogues have the fundamental floor that clarification dialogues are initiated by the system, based on internal error detection mechanisms. When a wrong interpretation passes all internal tests, the subject cannot do anything about it, actually he will not even notice it until the robot starts executing the instructions. A more natural interaction mode must be developed, where the user can probe the robot’s knowledge. This raises issues of knowledge representation and generation of verbal descriptions.
- Users are very adaptive and, given the appropriate prompts, would be happy to learn how to speak to the robot, just to make the communication more effective. Thereby, some of the problems we faced when dealing with unconstrained spoken language could be eliminated. This however requires a smarter approach to clarification dialogues, e.g. developing methods effective in “driving“ users towards using expressions understandable by the system.
- We were able to design robust primitives in our simplified environment, thereby demonstrating that underspecification can be handled. This is an important result supporting the viability of the IBL concept. Indeed, problems will be much more difficult when dealing with real-world scenes. However, we feel that some elements of the template-based approach developed here could be transposed in the real-world.

Some of the problems identified here will be addressed in a follow-up grant application.

3. Dissemination and research impact

Publications:

- 2 journal papers (*IEEE Intelligent Systems, Robotics and Autonomous Systems*). 1 Short paper in *AISB Quarterly*. 1 paper submitted (*Robotics and Autonomous Systems*). 1 overview paper planned.
- 1 Thesis
- 8 conference presentations (6 orals, 2 posters)

Project web page with publications, video footages, downloadable software and corpus data.(see IBL-Web)

Invited presentations (GB)

- Seminar at Derriford Hospital, 9 Feb 2000, “Robot Instruction by NL”.
- Seminar at ENSEA, Paris, 11 March 2000, “Robot Instruction by NL”

- Presentation at the European Robotics Research Meeting, Las Palmas, 6 January 2001 "Instruction-Based Learning for Mobile Robots"
- BCS SGES Evening Lectures, Birkbeck College, London, 25 April 2001 " Programming Robots with NL"
- Presentation at the European Robotics Research Meeting, Pisa, 2 February 2002 "Progress in Instruction-Based Learning for Mobile Robots"
- Seminar at the University of Essex, 8 February 2002, "Instructing Robots using NL"
- Invited speaker at the International Workshop on Multimodal Communication, 9-11 January 2003, Bielefeld Germany, “Design of a system for robot instruction using NL”
- Seminar at Aberystwyth, 10 December 2003 “Corpus-Based Robotics: A route instruction example”

Research visit: ENSEA, Paris, Image and Signal Processing Group, 17-28 June 2001.

Demonstration: Weird World Interactive Science Exhibition, City Museum Plymouth, 4-8 March 2003. (IBL system experimented with by Lord Sainsbury. Videos available on the IBL webpage)

4. Management and finances

Management: The project employed a PhD student (Theocharis Kyriacou) and a PostDoc (Dr. Stanislao Lauria). TK had completed a BSc in Electrical Engineering at Sheffield with a 2.1 and worked effectively on vision, primitives programming, and robot control. He has submitted a PhD on the design of robot primitives. TK will shortly start a postdoc in robotics at Essex. SL had done his PhD and a Post Doc at the Cybernetics Department at Reading. He dealt with the system architecture, the translation of semantic representations into calls to robot primitives, and learning. SL left 3 month before the end of the project to take a lectureship in computing at Brunel, where he is pursuing this area of research in collaboration with Plymouth. His early departure led to a 6 month extension of the project and the employment of Dr. Rohana Rajapakse for 2 month to perform the evaluation of the final system with a group of human subjects.

Finances. SL had to be employed on a scale two points higher than planned in the proposal. We used for that purpose funds originally planned for part-time secretarial support. The early departure of SL enabled to extend the studentship of TK by 4 months to provide support during final tests. The employment of RR was funded by transferring some travel money into the salary budget. The consumable budget appears to be overspent and the equipment budget underspent due to a number of pieces of equipment costing less than 1000 UKP being classified as “consumables”.

5. References

Bugmann G., Lauria S., Kyriacou T., Klein E., Bos J., Coventry K. (2001) Using Verbal Instructions for Route Learning: Instruction Analysis . Proc. TIMR 01 – Towards Intelligent Mobile Robots, Manchester 2001. Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 – 6161. Report number UMC-01-4-1.

Bugmann G. (2004) Corpus-Based Robotics: A route Instruction example. To appear in Proc. IAS-8, March 2004, Amsterdam.

Crangle C. and Suppes P. (1994) Language and Learning for Robots, CSLI Lecture notes No. 41, Centre for the Study of Language and Communication, Stanford, CA.

Gregor R., Lutzler M., Pellkofer M., Siedersbeger K.-H., Dickmanns E.D. (2002) EMS-Vision: A perceptual system for autonomous vehicles. IEEE Trans. Intel. Transp. Systems, 3:1, 48-59.

Huffman S.B. and Laird J.E. (1995) "Flexibly Instructable Agents", *Journal of Artificial Intelligence Research*, 3, pp. 271-324.

IBL-Web: www.tech.plym.ac.uk/soc/staff/guidbugm/ibl or follow links in www.tech.plym.ac.uk/iq4bots

Kyriacou T. (2004) Vision-based urban navigation procedures for verbally instructed robots. Thesis submitted. University of Plymouth.

Kyriacou T., Bugmann G., Lauria S. (2004) Vision-Based Urban Navigation Procedures for Verbally Instructed Robots. Submitted to Robotics and Autonomous Systems.

Lauria S., Bugmann G., Kyriacou T., Bos J., Klein E. (2001). "Personal Robot Training via Natural-Language Instructions. *IEEE Intelligent Systems*, 16:3, pp. 38-45.

Lauria S., Bugmann G., Kyriacou T., Klein E. (2002a) "Mobile Robot Programming Using Natural Language" *Robotics and Autonomous Systems*, 38 (3-4): 171-181

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

Matsui T., Asoh H., Fry J., Motomura Y., Asano F., Kurita T., Hara I. and Otsu N. (1999) Integrated Natural Spoken Dialogue System of Jijo-2 Mobile Robot for Office Services, Proc. AAAI/IAAI, pp. 621-627.

Schaal S., Ijspeert A.J. and Billard A. (2003) Computational approaches to motor learning by imitation. Phil. Trans. Roy. Soc.B: Biol. Sciences, 358, 537-547.

Torrance M.C. (1994) Natural Communication with Robots, MSc Thesis submitted to MIT Dept of Electrical Engin. and Comp. Science.