

Using Verbal Instructions for Route Learning: Instruction Analysis.

Guido Bugmann, Stanislao Lauria, Theocharis Kyriacou, Ewan Klein*, Johan Bos*, Kenny Coventry⁺

Centre for Neural and Adaptive Systems, School of Computing, University of Plymouth

⁺Department of Psychology, University of Plymouth,
Drake Circus, Plymouth PL4 8AA, United Kingdom

*Institute for Communicating and Collaborative Systems, Division of Informatics, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, United Kingdom

<http://www.tech.plym.ac.uk/soc/staff/guidbugm/ibl/index.html>

5/4/2001

Abstract

Future domestic robots will need to adapt to the special needs of their users and to their environment. It is likely that programming by natural language will be a key method enabling computer language-naïve users to instruct their robots. This paper describes initial steps and considerations towards the design of Instruction-Based Learning (IBL) systems. The proposed methodology is to be tested in the restricted domain of route instructions with real speech input and a real mobile robot using vision for navigation. Users will use unconstrained speech within a restricted domain-specific lexicon determined by analysing a corpus of route instructions. This will maximise speech recognition performance. The robot will possess an appropriate set of primitive procedures that correspond to procedures found in route instructions. Based on 96 route instructions, it is found that the task vocabulary contains approximately 270 words, but is not closed. It increases at an average rate of one new word for every new route instruction, although there are large inter-individual differences. It is also found that 58% of instructions contain no out-of-vocabulary words. The functional vocabulary is found to include 12 different procedures, and is also not closed. It increases at an average rate of one new procedure for every 25 instructions.

1. Introduction

Future domestic robots will be required to perform tasks that manufacturers cannot pre-program. For instance, making tea to the taste of the user, or fetching a book in a specified room. Such task require knowledge about the layout of the home of the user and on his/her preferences. Other tasks will need to be performed at a given time of the day or if certain conditions are met. In this paper we concentrate on navigation tasks, for instance the ones performed by an autonomous wheelchair carrying his/her user to a desired destination. The problem addressed is here the one of how a user, with no programming skills, could interact with the robot to modify its internal program.

The question of how robots could learn from their users has been investigated so far along two main routes, learning by imitation [Billard et al., 1998] and learning by reinforcement [e.g. Perez-Urbe and Hirsbrunner, 2000]. However, both methods have limited scope. For instance, learning by imitation does not enable the acquisition of rules such as "IF -THEN". Learning by reinforcement is a lengthy process that is best used for refining low-level motor control, but becomes impractical for complex tasks. Both methods do not readily generate knowledge representations that the user can interrogate. This paper focuses on another form of learning, by verbal instruction, that has proven its effectiveness in human learning [Bloom, 1984], but has received relatively little attention in robotics.

Previous work on verbal communication with robots has mainly focused on issuing *commands*, i.e. activating pre-programmed procedures using a limited vocabulary (e.g. IJCAI'95 office navigation contest). Only a few research groups have considered *learning*, i.e. the stable and reusable acquisition of new procedural knowledge. [Huffman & Laird, 1995] used textual input into a simulation of a manipulator with a discrete state and action space. [Crangle and Suppes, 1994] used voice input to teach displacements within a room and

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. [Http://www.cs.man.ac.uk/csonly/cstechrep/titles01.html](http://www.cs.man.ac.uk/csonly/cstechrep/titles01.html)

mathematical operations, but with no reusability. In [Torrance, 1995], textual input was used to build a graph representation of spatial knowledge. This system was brittle due to place recognition from odometric data and use of IR sensors for reactive motion control. Knowledge acquisition was concurrent with navigation, not prior to it.

Key criteria for the design of practical instruction-based learning (IBL) systems are seen here as: 1. Handling of natural speech, with its variations, underspecifications and errors in speech recognition; 2. Handling real world continuous state spaces with uncertainty and noise; 3. Incremental learning, with new instructions reusing previously taught procedures; 4. User-friendly and effective dialogue management by the robot.

Satisfying these criteria imposes numerous inter-linked constraints on the system architecture, the robot control design and the natural language processing component of an IBL system. In order to explore these effects, a simple route learning task has been selected, using real speech input and a robot using vision to execute the instructed route. The interaction scenario and the architecture of the proposed IBL system are outlined in section 2. Speech recognition, natural language understanding and dialogue management are described in section 3. The miniature experimental environment and the robot are described in section 4. As a first step toward designing a system that can handle unconstrained speech we have collected a corpus of data on unconstrained instructions given by users. The corpus collection procedure and the analysis of the data are described in section 5. Analysis is done along two lines, i) specifying the lexicon and ii) determining a list of primitives navigation procedures referred to in the instructions. The implications of the findings are discussed in section 6.

2. IBL concept: Interaction Scenario and Architecture.

2.1 Concept

The aim of the IBL project is to develop a system that converts verbal instructions into internal program code. Procedures learnt from the user become part of a pool of procedures that can be reused to learn more and more complex procedures. Hence, the robot becomes able to execute increasingly complex tasks.

To evaluate the potential and limitations of IBL, a real-world instructions task should be used, that is simple enough to be realisable, and generic enough to warrant conclusions that hold also for other task domains. To be generic, the task should require the learning of the three fundamental components of computer programs: Sequence, Selection and Repetition. These components are found in route instructions. First, a route is a sequence of route-segments. Secondly, although decisions are rarely part of route instructions (e.g. "if the road is blocked take this other one"), they are implicit in the execution of all segments. For instance, "take the first left" is to be translated in programming terms into "IF you are not at the intersection yet, THEN keep moving towards it. ELSE: do the left turn". Thirdly, explicit repetitions do occur in route instructions ("turn left 2 times"), and are also implicit in all segments which, as in the example above, requires a repetition of procedures ("keep moving until ...").

In terms of user-robot interaction, a typical learning process would start with the user asking the robot to perform a given task. If the robot lacked information about the task, it would ask for clarification or may ask the user to explain the task step by step. The ensuing dialogue constitutes the core of "instruction-based learning".

Due to the nature of the users, a requirement of the project is the use of unconstrained speech. In terms of vocabulary, this means that the user is allowed to use the words that are natural to him. However, using a restricted lexicon improves the performance of a speech recogniser. It is planned here to use a restricted lexicon that matches the one naturally used by users, so as to allow unconstrained speech. In terms of navigation procedures, the user is allowed to construct routes using functional primitives that are natural to him. It is planned to provide the robot with pre-programmed counterparts to these primitives, to enable a seamless conversion of verbal route instructions into programs. Corpus analysis along these lines is described in section 5. Another dimension of unconstrained speech is dialogue management. The user should be free to initiate or terminate dialogue moves at will. For instance, the user should be able at any time to interrupt a process in the robot, by issuing the command "stop", or leave a learning dialogue to issue a new command. This requires a flexible dialogue management but also a purpose-designed system architecture.

2.2 System Architecture

The architecture is comprised of several functional processing modules (figure 1). These are divided into two major units: the Dialogue Manager (DM) and the Robot Manager (RM).

The DM and the RM are designed as two different processes based on asynchronous communication protocols. These processes run concurrently on different processors. In this way, the system can handle, at the same time, both the dialogue aspects of an incoming request from the user (i.e. speech recognition and semantic analysis, or detection of a "stop" command) and the execution of a previous user request (i.e. check if the request is in the system knowledge domain, and execute vision-based navigation procedures).

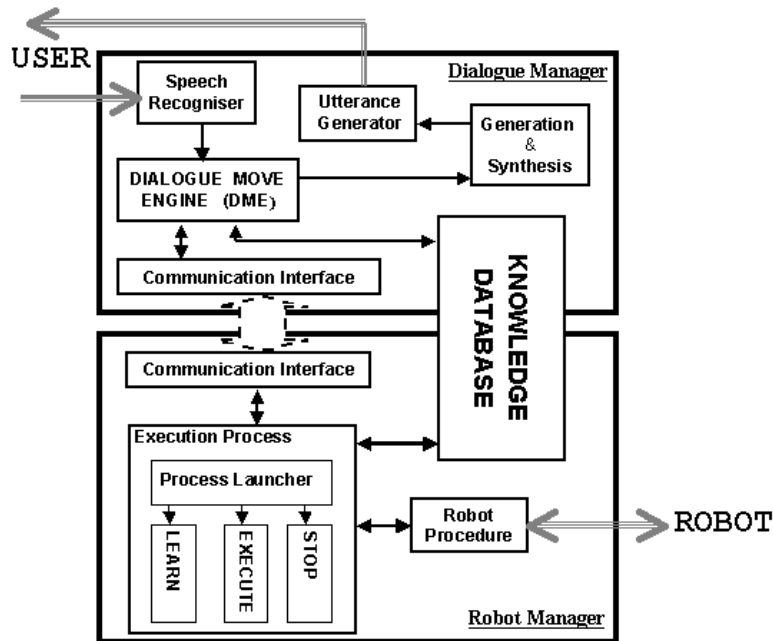


Figure 1. IBL systems architecture.

Two aspects are essential with this concurrent-processes approach. Firstly, to define an appropriate protocol between the two processes. Secondly, to define an appropriate architecture for the RM and DM allowing the two processes to both communicate with each other while performing other tasks. At present a communication protocol based on sockets and context-tagged messages is evaluated.

Moreover, the system must also dynamically adapt itself to new user requests or to new internal changes by being able to temporarily suspend or permanently interrupt some previous activity. For example the user may want to prevent the robot crashing against a wall and must therefore be able to stop the robot while the robot is driving towards the wall. Hence, the importance of a concurrent approach where the systems constantly listens to the user while performing other tasks and, at the same time, is able to adjust the task if necessary.

The Dialogue Manager is a bi-directional interface between the Robot Manager and the user, either converting speech input into a semantic representation, or converting requests from the Robot Manager into dialogues with the user. Its components run as different processes communicating with each other via a blackboard architecture.

The RM must concurrently listen/send requests from/to the DM and try to execute them. For this reason a multi-threads approach has been used. The communication interface is a process that only launches a message evaluation thread "Execution Process" and resumes listening to the DM. The execution process then starts an appropriate thread for executing a command, or places a tagged message on a message board if it is a part of a dialogue in a specific thread, e.g. learning a route. The characteristic of this approach is that all processes in the RM are sharing common memory stack so that threads can be started and paused, depending on the user's input.

The Robot Manager is written using the scripting language Python¹ and C. An important feature of scripting languages is their ability to write their own code. For instance, a route instruction given by the user will be saved by the Robot Manager as a Python script that then becomes part of the procedure set available to the robot for execution or for future learning.

¹ <http://www.python.org>

3 Natural Language processing and dialogue management.

3.1 The Dialogue Move Engine

The ongoing dialogue between user and robot is represented by a discourse representation structure (DRS) proposed by Discourse Representation Theory (Kamp & Reyle 1993). New utterances yield DRSs (see section 3.2 below), which update the DRS of the dialogue, following the recent information state approach to dialogue processing (Traum et al. 1999). Context-sensitive expressions (such as pronouns and presupposition triggers) are resolved with respect to this DRS. Finally, utterances of the robot are realized by generating prosodically annotated strings from DRSs and feeding these to a synthesizer.

Using semantic representations for modeling the dialogue is motivated by the need to perform inferences in order to let the robot make "intelligent" responses. Inferences are required to resolve ambiguities present in the user's input (being of scopal, referential, or lexical nature), to detect the speech act associated to an utterance (e.g., did the user answer a question or has a new issue been raised?), to plan the next utterance or action, and to generate naturally sounding utterances (e.g., by distinguishing old from new information within an utterance). Inference are actually carried out using off-the-shelf theorem provers, by translating DRSs to first-order logic (cf. Blackburn et al. 1999).

3.2. Speech Recognition

Speech recognition and semantic construction are integrated into one component. The basic idea is to use off-the-shelf speech recognition, and to use a grammar that is linguistically motivated and domain independent. The grammar not only consists of rules that determine the syntactic structure of utterances, but also features semantic rules that specify how semantic representations (underspecified DRSs) are built in a compositional way.

The current prototype implementation uses Nuance² tools for speech recognition. The initial grammar is a unification-based phrase structure grammar, which is compiled into GSL, the Grammar Specification Language supported by Nuance's technology. This compilation involves removing left-recursive rules within the grammar, as well as replacing features and their possible values for syntactic category symbols, as GSL neither support left-recursive rules nor a feature-value system. As a consequence, the language models for the speech recognition are huge, but still feasible for small lexicons (a few hundred words in the case of IBL).

The semantic operations are compiled-out in GSL as well, and each word in the lexicon is associated with a semantic representation. As a result, the output of the speech recognizer is directly a semantic representation, in our case an underspecified DRS, and another step of processing (such as parsing and semantic construction) is not required. Hence, by compiling our linguistic grammar into GSL, we short-cut the parsing and semantic construction process into a single component.

4 Experimental Environment and task.

The environment is a miniature town covering an area of size 170cm x 120cm (figure 2). The robot is a modified RobotFootball robot³ with a 8cm x 8cm base (figure 3A). The robot carries a CCD colour TV camera⁴ (628 (H) x 582 (V) pixels) and a TV VHF transmitter. Images are to be processed by a PC that acquires them via with a TV capture card⁵ (an example of such image is shown in figure 3B). The PC will then send motion commands by FM radio to the robot. During corpus collection, the PC is also used to record instructions given by subjects.

The advantage of a miniature environment is the ability to build a complex route structure in the limited space of a laboratory. The design is as realistic as possible, to enable subjects to use natural expressions for the outdoor real-size environment. Buildings have signs taken from real life to indicate given shops or utilities such as the post-office. However, the environment lacks some elements such as traffic lights that may normally be used in route instructions. Hence the collected corpus is likely to be more restricted than for outdoor route instructions.

The advantage of using a robot with a remote-brain architecture [Inaba et al., 2000] is that the robot does not require huge on-board computing and hence can be small, fitting the dimensions of the environment.

² <http://www.nuance.com>

³ Provided by Merlin Systems (<http://www.merlinsystemscorp.com/>)

⁴ Provided by Allthings Sales and Services (<http://www.allthings.com.au/>)

⁵ TV Card: Hauppauge WinTV GO



Figure 2. Miniature town in which a robot will navigate according to route instructions given by users.

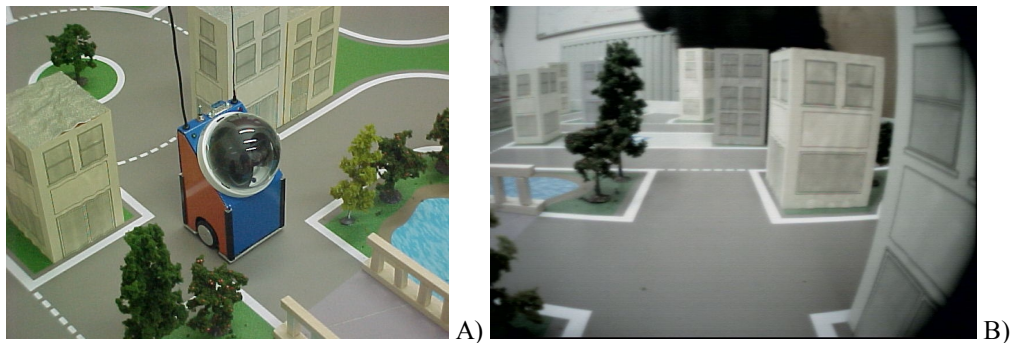


Figure 3. A. Miniature robot (base 8cm x 8cm). B. View from the on-board camera.

5 Corpus collection and data analysis

5.1 Data collection

To collect linguistic and functional data specific to route learning, 24 subjects were recorded as they gave route instructions to the robot in the environment. Subjects were divided into three groups of 8. The first two groups (A and B) were told that the robot was remote-controlled and that, at a later date, a human operator would use their instructions to drive the robot to its destination. It was specified that the human operator would be located in another room, seeing only the image from the wireless on-board video camera. This was specified to induce the subject into using spatial references accessible by the future vision software. Subjects were also told to reuse previously defined routes whenever possible, instead of re-explaining them in detail. Each subject had 6 routes to describe among which 3 were “short” and 3 were “long”. Each long routes included a short one, so that users could refer to the short one when describing the long one, instead of re-describing all segments of the short one. This was to reveal the type of expressions used by users to link taught procedures with primitive ones. Each subject described 6 routes having the same starting point and six different destinations. Starting points were changed after every two subjects. Thus a total of 48 route descriptions were collected from each group. Groups A and B received the same routes to describe, but with the sequence of “short” and “long” routes inverted. This would reveal the difference between a fully detailed route, and a route with reference to a short route inserted. Again the question is the one of how procedure insertion is handled by subjects. The first two groups (A and B) used totally unconstrained speech, to provide a performance baseline. It is assumed that a robot that can understand these instructions as well as a human operator would represent the ideal standard. A third group of 8 subjects (C) had the same routes to describe as group A, but were forced into a simplified dialogue with an operator to produce shorter chunks of descriptions. It is known that it is very difficult for NL

processing tools to correctly segment an uninterrupted stream of words into sentences. Therefore, corpus C may be more representative of utterances in the eventual user-robot dialogue. Subjects in this group were told that the operator next door was taking notes. A researcher pretended to do so and interrupted the subjects (using a microphone) when they used chunks that were too long. He acted as if he understood all the instructions and did not initiate repair dialogues. The analysis performed so far covers group A and group C. Table 1 shows an example of the same two “short” and “long” routes instructed by a subject in group A and a subject in group C. The instructions were transcribed in XML using the Transcriber⁶ software.

Monologue	
Short	User: okay take your first right and continue down the street past Derry's past Safeway and your parking lot the car park will be on your right
Long	User: okay once you pass the car park er take your first right and then again take your first right and the hospital will be right in front of you
Dialogue	
Short	Wizard: <i>could you tell me how to get to the car park please</i> User: okay you'll take the first right from where you are now past Derry's then Safeway Wizard: <i>yes</i> User: you'll pass another road on the left and the car park's on the right from there Wizard: <i>thank you</i>
Long	Wizard: <i>could you tell me how to go to the hospital please</i> User: okay you need to go back towards the car park Wizard: <i>yes</i> User: past the car park take the first right Wizard: <i>i'm sorry after i pass the car park</i> User: you take the right after the car park Wizard: <i>yes</i> User: and then another right again Wizard: <i>yes</i> User: and you'll be moving towards the hospital on the end of that road Wizard: <i>thank you</i>

Table 1: Example of instructions for a short route from E to P and a long route from E to H (see figure 2) given under monologue condition (group A) and dialogue condition (group C). The wizard is a human operator mimicking verbal feedback that could be given by the robot.

5.2 Analysis of the Task Vocabulary

To provide an initial estimate of the task vocabulary, the data from group A and C were merged. The number of distinct words was counted in the set of 96 instructions given. Morphology was not taken into account, i.e. “travels” and “travel” were counted as different words. The vocabulary of the users was found to contain 269 different words, from a total of 4020 word in the combined corpus A and C. The most frequent words were found up to 491 times and 67 words were used only once (table 2), i.e. only one subject used a particular word in a single route instruction.

To determine if the corpus collection had led to a complete sampling of the task vocabulary, the average number of distinct words was plotted as a function of the number of collected instructions. Figure 4 shows that the number of distinct words is still rising at the end of the curve, indicating that more new words would be found if more route instructions were collected. This behaviour is similar in other task domains [Zue, 1997]. The slope of the curve in figure 4 indicates that a new user might say on average one out-of-vocabulary word in each instruction.

To determine what type of new word might be expected, each route instruction was compared to the corpus of all other instructions. The result is that the new words are all among the 69 least frequently used words. Table 2 shows that these are not necessarily "unusual" words. The question of how the understanding of an instruction might be affected by the absence of such words from the vocabulary will be investigated further.

The dialogue group tended to use less distinct words (figure 3) and tended to produce less "out-of-vocabulary" words (table 3). Therefore, future experiments may reveal an improved speech recognition performance in dialogue conditions.

⁶ <http://www ldc.upenn.edu/mirror/Transcriber/>

Most Frequent		Least frequent
Word	Count	
The	491	Order, here, doors, onto, robot,
And	166	well, center, moving, moment,
On	162	thank you, lot, park's, actually, its,
You	125	carrying, able, tesco, sharp,
To	123	turned, leave, arrive, branch,
Take	117	taking, while, crossing, hundred,
Left	114	taken, double, bears, area, ninety,
Right	108	instruct, turnings, feel, apologize,
Go	97	thirty, or, place, amount, leaving,
Your	92	time, blocks, diagonally, there's,
		say, currently, what, reaching,
		travels, some, bear, bends, says,
		means, quadrangle, exits, like,
		forty-five, set, now, half, five, very,
		only, uh-huh, certainly, tesco's,
		paper, quarters, soon, move

Table 2: Most frequent and least frequent user word in the corpus. The least frequent words were found only once in 96 route descriptions.

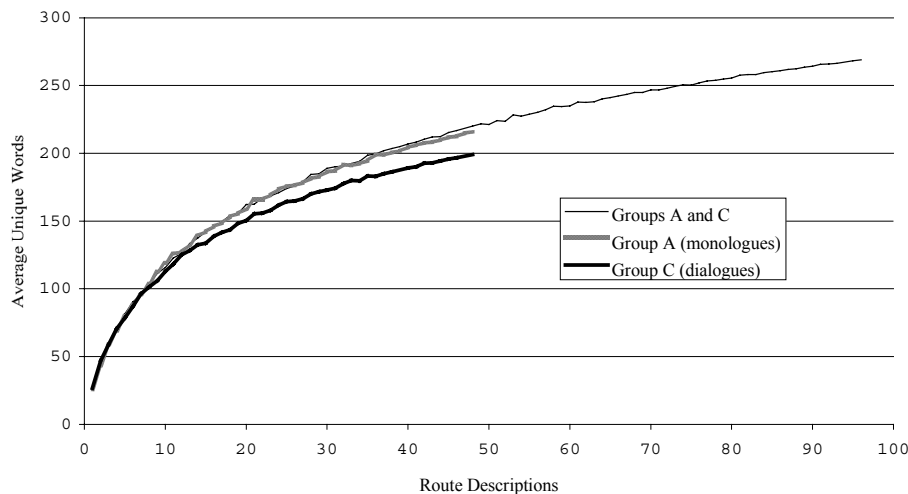


Figure 4. Number of distinct words discovered in the corpus as the number of instruction samples increases. The long line is for group A and C pooled. The shorter lines are for groups A and C taken in isolation. Curves are obtained by averaging over 48 random sets comprising an increasing number of sample instructions. The slope of the long curve indicates that, on average, one new word is added to the vocabulary for every additional route instruction collected.

Another way to look at the problem of out-of-vocabulary words, is to determine how many instructions actually contain new words. The result is that 58% of instructions had no new words (60% of those in group A, and 56% of those in group C) implying that more than half of the instructions would be recognised perfectly by a speech recognition system based on the current vocabulary. Among the remaining 42% of instructions, 65 % had only one new word and 35% had between 2 and 6 new words.

Subject:	2	3	4	5	6	7	8	9	1	10	11	12	13	14	15	16
Group:	d	d	d	d	d	d	d	d	m	m	m	m	m	m	m	m
Nb. New words	2	3	10	5	5	0	0	2	2	3	3	1	1	19	10	3

Table 3: Number of new words used by each subject in their 6 route instructions. The group of the subjects is indicated by d = dialogue (C) or m = monologue (A). There are 27 new words in group C and 42 in group A.

There was also a significant inter-subject variability (table 3). Some subjects used less than two new words in their 6 descriptions, while others were blessed with a particularly rich vocabulary and produced several new words in each one of their instruction. This is not necessarily a blessing when it comes to interacting with a robot. However, some of the “new” words counted here were morphological variations of known words, and a speech recognizer would have recognized them. In general, not more than one sentence per instruction contains a truly out-of-vocabulary word. Hence, it is expected that situations where repair is needed will not be unbearably frequent, but they are likely to affect most users.

5.3 Analysis of the Functional Vocabulary.

The functional vocabulary is a list of primitive navigation procedures found in descriptions. The initial annotation of instructions in terms or procedures, as reported here, is somehow subjective, and influenced by two considerations. 1. The defined primitives will eventually be produced as C-Programs. It was hoped that only a few generic procedures would have to be written. Therefore, the corpus has been transcribed into rather general procedures characterised by several parameters (table 4). 2. An important issue is knowledge representation. A route is to be represented as a graph, constituted of a continuous chain of primitives. For that purpose, all primitives must be consistent with a standard " $S_i A_{ij} S_j$ " representation (Initial state S_i , final state S_j and linking action A_{ij}). For a route description to be accepted as complete and executable, the initial state of each procedure must correspond to the final state of the previous one.

Subjects however rarely specified explicitly the starting point and it was assumed that the system would need to be able to infer the starting point from previous action specifications. Therefore, procedures without starting points were considered complete, and were annotated as such. The specifications of primitive procedures is likely to evolve during the project.

This methodology differs from the one used in [Denis, 1997]. Denis converted each instruction into a propositional format. For instance "You will arrive at a wooden bridge that you must cross" is converted into:

1. ARRIVE AT(YOU, BRIDGE);
2. WOODEN(BRIDGE);
3. CROSS(YOU, BRIDGE)

Statements in this format were grouped into four classes: "prescribing action" (e.g. "turn left"), "prescribing actions with reference to a landmark" (e.g. number 3 above), "introducing landmarks" (e.g. "there is a tree to your left"), "describing landmarks" (e.g. number 2 above) and "Commentaries" (e.g. "the route will take about 5 min.").

In our analysis, there are no statements describing landmarks, as these are included in the termination points and there are no actions without reference to landmarks, as robot procedures need a defined termination point. Even when a subject specified a non-terminated action, such as "keep going", it was classified as "MOVE FORWARD UNTIL", assuming that a termination point would be inferred from the next specified action. The list of actions found in the descriptions of groups A and C is given in table 4.

	Count	Primitive Procedures
1	178	MOVE FORWARD UNTIL [(past over across) <landmark>] [(half_way_of end_of) street] [after <number><landmark> [left right]] [road_bend]
2	118	TAKE THE [<number>] turn [(left right)] [(before after at) <landmark>]
3	94	<landmark> IS LOCATED [left right ahead] [(at next_to left_of right_of in_front_of past behind on opposite near) <landmark >] [(half_way_of end_of beginning_of across) street] [between <landmark> and <landmark>] [on <number> turning (left right)]
4	49	GO (before after to) <landmark>
5	32	GO ROUND ROUNDABOUT [left right] [(after before at) <landmark>]
6	27	TAKE THE <number> EXIT [(before after at) <landmark>]
7	9	FOLLOW KNOWN ROUTE TO <landmark> UNTIL (before after at) <landmark>
8	3	STATIONARY TURN [left right around] [at from <landmark>]
9	1	TAKE THE ROAD in_front
10	1	PARK AT <location>
11	1	CROSS ROAD
12	1	EXIT [car_park park]

Table 4. Primitive navigation procedures found in the route descriptions collected from groups A and C. Procedure 3 is used by most subjects to indicate the last leg of the route, when the goal is in sight.

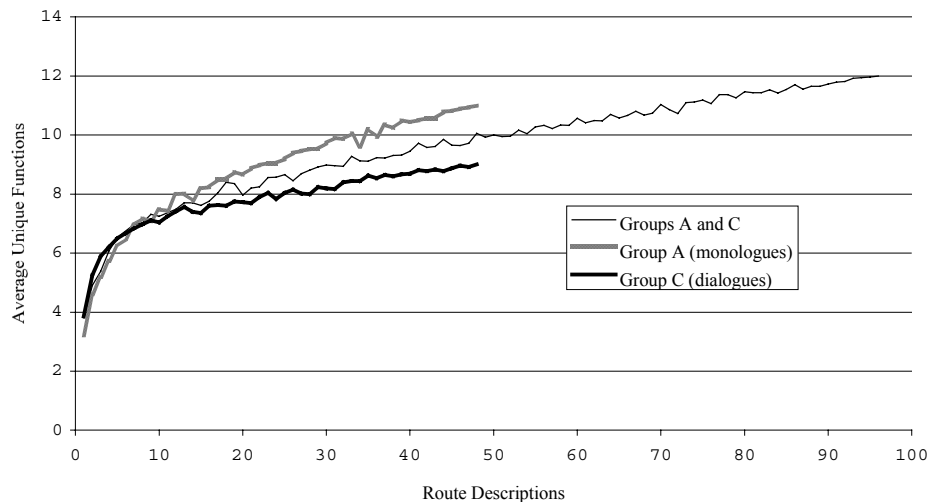


Figure 5. Average number of unique procedures as a function of the number of collected route instructions (Curves calculated as in figure 4.).

Figure 5 shows that the number of distinct procedures is increasing with the number of sampled instructions, but at a rate much smaller than the number of distinct words seen in the previous section. Here we discover on average one new procedure for every 25 route instructions, while with words, we discovered in average one new word for each instruction (figure 4). New procedures typically are the least frequent in table 4.

6 Discussion

Teaching a route to a robot using natural language is only one application of a more general instruction-based learning methodology. The approach described here aims at providing users with the possibility of using unconstrained speech, whilst creating an efficient natural language processing system using a restricted lexicon. The preliminary analysis of the lexicon shows however that out-of-vocabulary errors are to be expected. This is a well known problem in the domain of speech recognition, but it is a rather new observation on the functional side. From a roboticist's point of view, route navigation can be achieved with a rather small number of primitives. However, in spontaneous speech, a wider variety of functions must be expected.

We are attempting to give the user the freedom to reply or not to reply to a query, to control when given dialogues are to take place and to interrupt the robot at will. This created interesting constraints on the design of the system's architecture. In particular it calls for a solution using multi-threads with shared memory. Experiments will reveal how effective this solution is.

The results in section 5.2 indicate that when working with a limited vocabulary, it is unavoidable that unknown words are going to be used by users. This is the price to pay for having a reasonably robust speaker independent recogniser. In current speech recognition systems, such words would either be ignored or replaced with the most likely word in the lexicon. Limited research has gone into speech recognisers that would signal that some sound is likely to be a new word and learn the new word [Zue, 1997; Asadi et al, 1991]. When working with large vocabularies, out-of-vocabulary words are less likely to occur, but word recognition errors then occur due to the larger search space. Thus in any case, error spotting and repair mechanisms need to be built into an IBL system.

Word recognition errors can be revealed in the DM when they cause ungrammatical sentences. The RM can also detect word errors when they lead to unknown tasks being requested. The last stage of error spotting is to ask the user to confirm a task just before execution. Overall, error spotting and repair is not a simple problem, and experiments will be needed to understand how best to approach it.

The functional vocabulary is rather small. It includes navigation procedures and cognitive⁷ procedures. An important finding is that the functional vocabulary is not closed. Hence, at some point in the robot's life, the

⁷ "cognitive" denotes here actions that manipulate knowledge as opposed to actions that move the robot.

user will have to teach it new primitives (e.g. "cross the road"). Future work will have to determine what additional set of primitives are needed by the robot to understand instructions explaining e.g. how to "cross the road". Another issue is the identification of new functions, as the lexicon may not contain the required words.

7 Conclusion

The project described in this paper is aimed at exploring IBL for a limited class of functions: routes descriptions. Hence steps were taken to pre-program all other functions necessary for constructing route descriptions. A corpus of instructions was analysed to determine the list of words that the speech recognition system should recognise. Similarly a list of primitive procedures was established to ensure that the robot would be able to execute the navigation procedures forming the instructions. However, the initial results presented here show that neither the lexicon nor primitive procedures are likely to form closed sets. Ideally, an IBL system should therefore also be capable of acquiring new words, and users should be given the possibility to teach new primitive procedures. Unfortunately, the former is beyond the capabilities of current speech recognition systems. As for learning new primitive procedures, this would require a new set of more primitive procedures to be combined via user instructions. Whether it will be possible to explore this during the project is unclear. To allow IBL to operate despite these limitations, it is likely that a crucial role will be played by dialogue management.

Acknowledgement: This work is supported by EPSRC grants GR/M90023 and GR/M90160.

References:

- Asadi A., Schwartz R. and Makhoul J. (1991) "Automatic modelling for adding new words to a large vocabulary continuous speech recognition system", Proc. ICASSP, pp. 305-308.
- Billard A., Dautenham K. and Hayes G. (1998) "Experiments on human-robot communication with Robota, an imitative learning and communication doll robot", Contribution to Workshop "Socially Situated Intelligence" at SAB98 conference, Zurich, Technical Report of Centre for Policy Modelling, Manchester Metropolitan University, CPM-98-38. (<http://www.cpm.mmu.ac.uk:80/cpmrep38.html>)
- Blackburn P., Bos J., Kohlhase M. and de Nivelde H. (1999). Inference and Computational Semantics. In: Third International Workshop on Computational Semantics (IWCS-3), Tilburg, The Netherlands.
- Bloom B.S. (1984) "The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring", *Education Researcher*, 13:6, pp. 4-16.
- 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.
- Denis M. (1997) "The description of routes: A cognitive approach to the production of spatial discourse", CPC, 16:4, pp.409-458.
- Huffman S.B. and Laird J.E. (1995) "Flexibly instructable agents", *Journal of Artificial Intelligence Research*, 3, pp.271-324.
- Inaba M., Kagami S., Kanehiro F., Hoshino Y., Inoue H. (2000) "A platform for robotics research based on the remote-brained robot approach", *International Journal of Robotics Research*, 19:10, pp. 933-954.
- Kamp H. and Reyle U. (1993): *From Discourse to Logic*. Kluwer.
- Perez-Urbe A. and Hirsbrunner B., "Learning and Foraging in Robot-bees", SAB2000 Proceedings Supplement Book, Meyer, Berthoz, Floreano, Roitblat and Wilson (Eds), Published by International Society for Adaptive Behavior, Honolulu (to appear). <http://www-iiuf.unifr.ch/~aperezu/robot-bees/> also <http://www-iiuf.unifr.ch/~aperezu/robotreinfo.html>
- PYTHON: <http://www.python.org>
- Torrance M.C. (1994) Natural Communication with Robots, MSc Thesis submitted to MIT Department of Electrical Engineering and Computer Science, January 28, 1994.
- Traum, D., J. Bos, R. Cooper, S. Larsson, I. Lewin, C. Matheson and M. Poesio (1999): A model of dialogue moves and information state revision. Trindi Report D2.1. Available from <http://www.ling.gu.se/research/projects/trindi>
- Young S.J. (2000) "Probabilistic Methods in Spoken Dialogue Systems", *Phil. Trans. Royal Society A*, 358: 1769, pp. 1389-1401 (<http://citeseer.nj.nec.com/386391.html>)
- Zue, V. (1997) "Conversational interfaces: Advances and challenges", In Proc. Eurospeech, pages 9-14, Rhodes, Greece. (<http://citeseer.nj.nec.com/78849.html>).