Corpus-Based Robotics: A Route Instruction Example.

Guido Bugmann, Ewan Klein*, Stanislao Lauria**, and Theocharis Kyriacou

Robotic Intelligence Laboratory, School of Computing, Communication and Electronics, 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, UK.
** Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex UB8 3PH, United Kingdom

Abstract: In corpus-based robotics, the primitive functions built into a robot are determined by the functional content of human utterances spoken to the robot. In the example of route instructions treated in this paper, the 15 primitives found include functions such as turn(), cross(), landmark_is_located(), etc. These are natural primitives of human behaviour but complex robot functions, some of which are not normally thought of by roboticists. Primitives must cope robustly with a variety of environmental conditions and require autonomous navigation capabilities based for instance on visual landmark recognition and localisation, navigable space mapping and path planning. Thus, the requirement of human-robot interaction creates specific and demanding functional targets for robot designers. A major obstacle to human-robot communication lies probably in current robots' perception capabilities. Furthermore, for human-robot communication to match the performance of human-human communication, the robot must also be provided with capabilities of re-interpreting and correcting instructions at execution time. Such a large autonomy raises wider issues of safety and control.

1. Introduction

Domestic or service robots are multi-functional and cannot be pre-programmed in factory for all possible tasks that the final user may think of. The question is thus, how will such robots be programmed or re-programmed during their life-time? In industrial robotics, robots are programmed infrequently by a small number of skilled operators. This model is not cost-effective in service robotics and may constitute a major bottleneck in the commercial development of the domain. Ideally, users themselves should be able to program their robots, without the need for learning specialized programming languages, control theory, artificial vision, AI, etc. Instead, a service robot must understand instructions issued in natural language and generate the corresponding internal machine code program ("Instruction-Based Learning").

Comparatively little research has been devoted to Instruction-Based Learning (IBL) [1],[2],[3],[4]. All these previous approaches used a constrained language that the user had to learn. Thus, the aim of the Plymouth-Edinburgh IBL project on which this paper is based was to develop a system that accepts utterances that are natural to the user. The focus of this paper however is not on the natural language processing or the learning aspects of the project, but on the implication for robotics of the use of unconstrained language.

Essentially, if the robot has to understand the spoken words "take the first turn left" it also needs to have the sensori-motor capabilities to execute the command. Thus, there is a direct link between natural language understanding capabilities and the functional specification of the robot. The robot design process must therefore start with a functional analysis of a representative corpus of utterances (commands or instructions) spoken by potential users. In this paper, such a corpus-based methodology and its consequences are discussed.

The paper is organized as follows. In section 2, the selected application domain and the corpus collection process is described. In section 3 the corpus analysis method is detailed. In section 4, the system development and evaluation methods are described. In section 5, experimental results are discussed. In section 6, the paper concludes with a summary of the main lessons learnt.

2. Corpus collection and application domain.

In this IBL project, a robot is instructed on how to travel from one place to another in a miniature town. On the basis of the user's instructions, it first creates a computer program script that it then uses to navigate between the two places. The route instruction domain was selected because of the familiarity of the authors with vision-based robot navigation problems and because of the generic character of route instructions. These are expected to contain all the main structures found in computer programs: selection, sequence and repetition. Thus, it is expected that the results obtained would generalize to other domains.

A miniature urban environment was built on a 170cm x 120cm area suitable for navigation by a miniature 8cm x 8cm robot. Subjects were placed in front of the town and were asked to instruct the robot (figure 1). The sessions were filmed and the utterances were recorded digitally with a good quality headset microphone.

Figure 1. A subject instructing the robot during corpus collection. Inset: remote-brained miniature robot.

Subjects were explained that the recordings would be used later by a human operator driving the robot by remote control by using only the images acquired from the on-board camera. This was aimed at generating robot-centric spatial references that would be appropriate for use by a computer program controlling the robot using only information from the on-board camera. By specifying that the user of the instructions would be a human, we obviated the

need for users to second-guess what a robot might be able to understand and avoided any distortion of their natural expressions.

We were also interested in how subjects would refer to route previously instructed to the robot. This is because, in principle, when a given procedure has been explained to a robot, a user should be able to re-use it and refer to it when explaining more complex procedures. Routes given to subjects were organized so that certain routes would be extensions of previous routes with a few more turns or intersections. Subjects were instructed that, when appropriate, they could save time by referring to previously explained routes, instead of re-explaining all of the steps.

3. Corpus analysis

The corpus contained 144 routes produced by 24 subjects instructing 6 routes each. Its analysis had two purposes: i) Customise the speech recognition and natural language processing system for the domain of the application and ii) Define the functional primitives of the robot.

On the natural language side, the recorded instructions were first chunked automatically into shorter utterances by detecting naturally occurring silences between functional chunks. This produced meaningful chunks in most of the cases. The utterances sound files were then transcribed by hand into text files. The transcripts were used to generate automatically a restricted grammar comprising the subset of all the rules of a wide-coverage grammar hit by the corpus. This restricted grammar is then used to specialize a speech recognition system for the route instructions domain [5][6].

On the robot primitive side, the corpus transcripts were annotated by hand, as there is no off-the-shelf tool for doing it automatically. The annotation consisted of identifying basic actions in the user instructions and then assign executable robot primitives to the basic actions (e.g. table 1.).

Table 1. Example of functional annotation of a transcription.

```
[take your first right]
→ TURN (first, right)
[continue down the street past derry's past safeway]
→ FOLLOW_ROAD_UNTIL (past, derry's)
→ FOLLOW_ROAD_UNTIL (past, safeway)
[the car park will be on your right]
→ DESTINATION  IS(car park, right)
```

This annotation of instructions in terms of primitive procedures is a somehow subjective process guided by the knowledge that formal programs would have to be written for the procedures. Thus there was a bias towards grouping actions under a number of procedures as small as possible. The consequence was that each primitive accepts a number of parameter combinations. E.g "turn left, turn right, take the second right, turn left after the church" are all grouped as one primitive "turn(p1, p2, p3, p4, …)", where p1,p2,.. are parameters such as *direction*, *ordinal*, etc. It should be noted that during the project, the specifications of primitives slightly changed due to various constraints. Another bias came from the need for a termination condition for each primitive. For instance when subjects say "keep going", this is a non-terminated action that would set the robot into an infinite loop. It turned out that such instructions can be ignored. The reason is that each of the other

terminating functions has a "keep going" function already built into it. For instance, "turn left" functionally means "keep moving until you reach the left turn, and then take the turn".

This analysis methodology differs slightly from the one in [7] performed as part of research in psychology. For instance, in our analysis, there are no statements describing landmarks, as these are integrated into the parameter list of the primitives, and consequently there are no actions without reference to landmarks either. The 15 primitives found in the development corpus are shown in table 2. For more details about their implementation see [8]. Note that some of these functions would probably not have been conceive of naturally by a roboticist without the help of the corpus, e.g. "cross_to, take_road, goto_side".

Table 2. List of primitives.

	Primitive	Description
1	**Go**(description_1, landmark_1, preposition_1, description_2, landmark_2)	Instructs the robot to follow a known route (with known starting point and destination).
2	**location_is**(description_1, landmark_1, direction_1, preposition_1, description_2, landmark_2, description_3, landmark_3, ordinal_1)	Specifies the location of a landmark along a route.
3	**destination_is**(description_1, landmark_1, direction_1, , preposition_1, description_2, landmark_2, description_3, landmark_3, ordinal_1)	Indicates the location of the destination landmark.
4	**go_until**(description_1, landmark_1, preposition_1, description_2, landmark_2)	Follow known route to a landmark until a specified location along the route.
5	**Exit_roundabout**(ordinal_1, preposition_1, description_1, landmark_1)	Take a specified exit from a roundabout.
6	**Turn**(ordinal_1, direction_1, preposition_1, description_1, landmark_1)	Take a specified turn off a road.
7	**follow_road_until**(preposition_1, description_1, landmark_1)	Move forward until a certain location.
8	**rotate**(direction_1, extend_1, around_1)	Rotate on the spot.
9	**Exit_from**(description_1, landmark_1)	Exit from a place, usually used for the car park.
10	**cross_to**(description_1, landmark_1)	Instructs the robot to cross the road to a landmark.
11	**enter_roundabout**(direction_1)	Enter the roundabout in a specific direction.
12	**park**(preposition_1, description_1, landmark_1)	Park on, or close, to a certain landmark.
13	**Take_road**(preposition_1, description_1, landmark_1)	Take a road in view.
14	**Goto_side**(preposition_1, description_1, landmark_1)	Go round a landmark to one of its sides.
15	**Fork**(direction_1)	Follow a one of the two branches of a fork (Y split).

4. Development and evaluation methods.

The corpus was divided at random into two sets of 72 route instructions: the development and the evaluation corpus. The development corpus was used to develop the NLP

components and to establish the list of primitives as described in section 3. The evaluation corpus was used for the evaluation of whole and parts of the developed IBL system.

The system comprises several computational stages (speech recognition, semantic analysis, procedure extraction, route learning, route following) and, as far as possible, these were evaluated individually as a well as a whole. The speech recognition stage was evaluated automatically by comparing its output with the text transcription of the corpus. The semantic analysis stage produces Discourse Representation Structures (DRS) that can not be evaluated automatically, as it was impractical to produce target DRSs from the corpus. In the Robot Manager (RM) the DRSs are translated into function calls by a purpose-designed translation routine using a Procedure Specification Language (PSL) to specify translation rules. The output of that stage could be compared with a list of function calls produced by hand for each utterance in the corpus. However, due to the variety of possible errors, it turned to be a problem in itself to conceive of a meaningful measure of performance. For instance, primitive calls could be missing, or in the wrong order, or with a wrong parameter. Eventually, errors were examined and classified by hand.

For each of the primitives identified in the corpus, a corresponding program had to be written. More details on the vision-based navigation aspects can be found in [8]. The main challenge was to create primitives that would cope robustly with wide variations in environmental conditions. This is a crucial requirement human-robot communication, as natural language references to actions are heavily under-specified. To maximize robustness, the primitives were tested extensively on the routes in the development corpus. Their final performance was evaluated using hand-written[1] script programs based on instructions in the evaluation corpus containing situations not seen during development.

The system as a whole was tested in two ways. Firstly, by using as input the recorded instructions in the evaluation corpus. Secondly, through tests with human subjects who engage in an instruction dialogue with the system.

To compare the system's instruction following performance with that of humans learners, human subjects were asked to listen to the instructions in the corpus, make notes and then drive the robot via remote control, using only images transmitted from the on-board camera.

5. Results and discussion

Corpus analysis. The corpus of 6600 words contained approximately 330 distinct words but was not closed. Data analysis showed that one new word was expected to appear for every two route instructions [9]. This is a general problem with corpora in a limited application domain. In the navigation domain, such "out of grammar" expressions are expected to consist for instance of personal names or landmark names. How to handle these cases properly is important for practical systems but is a yet unsolved problem. Future robots should to be able to learn new words and extend their grammar to cover their use.

We found 15 primitives, but again there were indications that the corpus is not closed, with a reference to a new function expected for every 35 instructions [10]. 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. [11]) but this is probably not a general solution.

---

[1] This enabled testing primitives independently of the performance of the other components of the system.

We also found that the route-instruction domain is not generic in the computation sense, as almost all instruction only comprised sequences. There were many cases of implicit decision and loops. For instance, a function such as "turn left" can be decomposed into "keep moving until a left turn is seen, then take the turn". However, there were no explicit references to IF-THEN-ELSE structures in the corpus. Another domain may be more appropriate to explicitly generate conditions and loops in instructions.

Speech recognition. Using standard error measures on the recognition of the recorded utterances, we obtained a 40% word error rate. However, a limited test with the final dialogue system where one of us 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 repeat utterances that were not correctly translated, based on internal verification procedures. So, it is possible that with a minor increase in the linguistic discipline of the users, speech recognition can become quite effective.

Detecting references to previous routes. During corpus collection, subjects were encouraged to refer to previously taught routes. It turned out that such references are very difficult to detect in instructions. 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 is just in front of the robot or a roundabout further away that can 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 could not be properly translated.

Interestingly, experiments with human subjects listening to the instructions showed that only 55% of references to previous routes were detected in the instruction. Only when subjects started to drive the robot did they notice that there was a problem.

Using references to previous routes when creating program codes. Almost all references to previous routes required only a partial use of the 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 proposal 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 [12]. 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 computational architecture of the robot controller.

Programming the final instruction. The final instruction of a route instruction is often a statement like "and you will see it there on your left"[2]. The final instruction is especially interesting as it 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 undertaken the difficult task of detecting the building,

---

[2] That is why there is no "enter_into" primitive corresponding to the primitive 9 in table 2 ("exit from"). Subjects just say "and the car park is in front of you"

identifying it from its sign and locating its entrance. Instead, a coloured strip was placed at the foot of the building to signal its position. In a real urban environment the final instruction would pose vision and control challenges that are at the limits of current technical capabilities.

Comparing human and robot performance. When the robot followed the programs produced by hand for the evaluation corpus, it succeeded in reaching the goal only in 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. Sometimes they referred to a crossroad as a T-junction. In some cases they used utterances like "pass the first intersection. At the second intersection turn left". If translated literally, the "second" intersection means two more after the previous one… Interestingly, in 3% of the routes, the robot failed because the subject referred to a parameter combination that was not encountered in the development corpus ("cross the car park"), or he 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, with the same 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, human 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. If this is the case, human-robot communication can 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.

Performance of internal error detection. Details of the error detection in the IBL system will be described elsewhere, due to lack of space. The main finding is that not all errors done by the system in translating instructions into a program are detected. For instance, 32% of the recorded instructions are converted into a program. However, only 14% of these programs would actually lead the robot to the goal. In 18% of routes, errors in the translation from speech to program were not detected. This is an important issue because it is not safe for a robot to start a task that is not correctly specified. A human-in-the-loop approach seems also required here.

6. Conclusion

The starting point of corpus-based robotics is the fact that users are naïve in robot programming concepts and can only use their own terms to explain a task. The consequence is that a robotic system needs to be designed around utterances and functions natural to the user.

Several complex issues arise from that approach. The first is that no domain is really closed and an effective system should be able to learn new words and new primitives. Secondly, very sophisticated sensory-motor capabilities are required from the robot, essentially because human speech is designed to communicate with beings having such

abilities. Thirdly, errors in instructions do not impair human-human communication as much as human-robot communication. Humans are able of re-interpreting erroneous instructions at execution time or modifying them. For a robot, this would imply a significant autonomy which raises questions of safety.

All in all however, given that future assistant robots will need to interact verbally with their users, it seems appropriate to use a corpus-based approach to robot design. It has the advantage of revealing early in the design process what bottleneck problems need to be solved.

Acknowledgements

References

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

[2] 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.

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

[4] 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.

[5] Bos J. (2002): Compilation of Unification Grammars with Compositional Semantics to Speech Recognition Packages. COLING 2002. Proceedings of the 19th International Conference on Computational Linguistics. Pages 106-112.

[6] Bos J., Klein E., Lemon O., Oka T. (2003) DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture. 4th SIGdial Workshop on Discourse and Dialogue, Sapporo, Japan.

[7] Denis M. (1997) "The description of routes: A cognitive approach to the production of spatial discourse", *Current Psychology of Cognition*, 16:4, pp.409-458.

[8] Kyriacou T., Bugmann G., Lauria S. (2002) Vision-Based Urban Navigation Procedures for Verbally Instructed Robots. Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS'02) EPFL, Lausanne, Switzerland • October 2002. pp.1326-1331.

[9] 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.

[10] 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.

[11] 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.

[12] Lauria S., Kyriacou T. Bugmann G., Bos J and Klein E. (2002) 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.