

Sentence Planning for Realtime Navigational Instructions

Laura Stoia and Donna K. Byron and
Darla Magdalene Shockley and Eric Fosler-Lussier

The Ohio State University
Computer Science and Engineering
2015 Neil Ave., Columbus, Ohio 43210
stoia|dbyron|shockley|fosler@cse.ohio-state.edu

Abstract

In the current work, we focus on systems that provide incremental directions and monitor the progress of mobile users following those directions. Such directions are based on dynamic quantities like the visibility of reference points and their distance from the user. An intelligent navigation assistant might take advantage of the user’s mobility within the setting to achieve communicative goals, for example, by repositioning him to a point from which a description of the target is easier to produce. Calculating spatial variables over a corpus of human-human data developed for this study, we trained a classifier to detect contexts in which a target object can be felicitously described. Our algorithm matched the human subjects with 86% precision.

1 Introduction and Related Work

Dialog agents have been developed for a variety of navigation domains such as in-car driving directions (Dale et al., 2003), tourist information portals (Johnston et al., 2002) and pedestrian navigation (Muller, 2002). In all these applications, the human partner receives navigation instructions from a system. For these domains, contextual features of the physical setting must be taken into account for the agent to communicate successfully.

In dialog systems, one misunderstanding can often lead to additional errors (Moratz and Tenbrink, 2003), so the system must strategically choose instructions and referring expressions that can be clearly understood by the user. Human cognition studies have found that the *in front of/behind* axis

is easier to perceive than other relations (Bryant et al., 1992). In navigation tasks, this suggests that describing an object when it is *in front of* the follower is preferable to using other spatial relations. Studies on direction-giving language have found that speakers interleave repositioning commands (e.g. “Turn right 90 degrees”) designating objects of interest (e.g. “See that chair?”) and action commands (e.g. “Keep going”)(Tversky and Lee, 1999). The content planner of a spoken dialog system must decide which of these dialog moves to produce at each turn.

A route plan is a linked list of arcs between nodes representing locations and decision-points in the world. A direction-giving agent must perform several content-planning and surface realization steps, one of which is to decide how much of the route to describe to the user at once (Dale et al., 2003). Thus, the system selects the next target destination and must describe it to the user. In an interactive system, the generation agent must not only decide what to say to the user but also when to say it.

2 Dialog Collection Procedure

Our task setup employs a virtual-reality (VR) world in which one partner, the direction-follower (DF), moves about in the world to perform a series of tasks, such as pushing buttons to re-arrange objects in the room, picking up items, etc. The partners communicated through headset microphones. The simulated world was presented from first-person perspective on a desk-top computer monitor. The DF has no knowledge of the world map or tasks.

His partner, the direction-giver (DG), has a paper 2D map of the world and a list of tasks to complete. During the task, the DG has instant feedback about

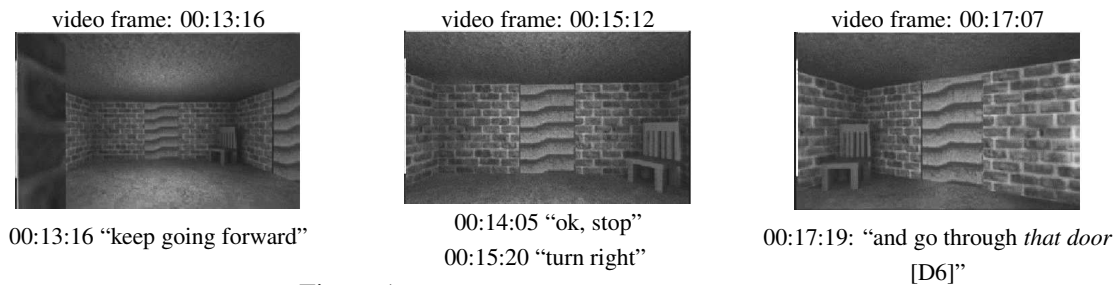


Figure 1: An example sequence with repositioning

DG: ok, yeah, go through *that door* [D9, locate]
turn to your right
 'mkay, and there's *a door* [D11, vague]
in there um, go through *the one*
straight in front of you [D11, locate]
 ok, stop... and then **turn around and look at**
the buttons [B18,B20,B21]
 ok, you wanna push *the button that's there*
on the left by the door [B18]
 ok, and then go through *the door* [D10]
look to your left
 there, in *that cabinet there* [C6, locate]

Figure 2: Sample dialog fragment

the DF's location in the VR world, via mirroring of his partner's screen on his own computer monitor. The DF can change his position or orientation within the virtual world independently of the DG's directions, but since the DG knows the task, their collaboration is necessary. In this study, we are most interested in the behavior of the DG, since the algorithm we develop emulates this role. Our paid participants were recruited in pairs, and were self-identified native speakers of North American English.

The video output of DF's computer was captured to a camera, along with the audio stream from both microphones. A logfile created by the VR engine recorded the DF's coordinates, gaze angle, and the position of objects in the world. All 3 data sources were synchronized using calibration markers. A technical report is available (Byron, 2005) that describes the recording equipment and software used.

Figure 2 is a dialog fragment in which the DG steers his partner to a cabinet, using both a sequence of target objects and three additional repositioning commands (in bold) to adjust his partner's spatial relationship with the target.

2.1 Developing the Training Corpus

We recorded fifteen dialogs containing a total of 221 minutes of speech. The corpus was transcribed and word-aligned. The dialogs were further anno-

tated using the Anvil tool (Kipp, 2004) to create a set of target referring expressions. Because we are interested in the spatial properties of the referents of these target referring expressions, the items included in this experiment were restricted to objects with a defined spatial position (buttons, doors and cabinets). We excluded plural referring expressions, since their spatial properties are more complex, and also expressions annotated as *vague* or *abandoned*. Overall, the corpus contains 1736 markable items, of which 87 were annotated as vague, 84 abandoned and 228 sets.

We annotated each referring expression with a boolean feature called **Locate** that indicates whether the expression is the first one that allowed the follower to identify the object in the world, in other words, the point at which joint spatial reference was achieved. The kappa (Carletta, 1996) obtained on this feature was 0.93. There were 466 referring expressions in the 15-dialog corpus that were annotated TRUE for this feature.

The dataset used in the experiments is a consensus version on which both annotators agreed on the set of markables. Due to the constraints introduced by the task, referent annotation achieved almost perfect agreement. Annotators were allowed to look ahead in the dialog to assign the referent. The data used in the current study is only the DG's language.

3 Algorithm Development

The generation module receives as input a route plan produced by a planning module, composed of a list of graph nodes that represent the route. As each subsequent target on the list is selected, content planning considers the tuple of variables <ID, LOC> where ID is an identifier for the target and LOC is the DF's location (his Cartesian coordinates and orientation angle). Target ID's are always object id's to be visited in performing the task, such as a door

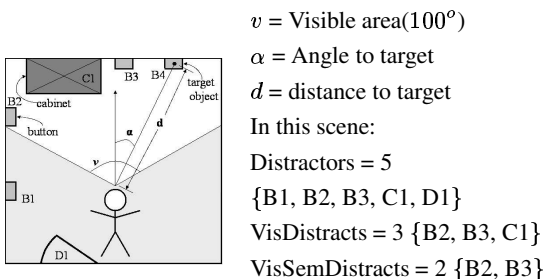


Figure 3: An example configuration with spatial context features. The target object is B4 and [B1, B2, B3, B4, C1, D1] are perceptually accessible.

that the DF must pass through. The VR world updates the value of LOC at a rate of 10 frames/sec. Using these variables, the content planner must decide whether the DF's current location is appropriate for producing a referring expression to describe the object.

The following features are calculated from this information: absolute **Angle** between target and follower's view direction, which implicitly gives the *in front* relation, **Distance** from target, visible distractors (**VisDistracts**), visible distractors of the same semantic category (**VisSemDistracts**), whether the target is visible (boolean **Visible**), and the target's semantic category (**Cat**: button/door/cabinet). Figure 3 is an example spatial configuration with these features identified.

3.1 Decision Tree Training

Training examples from the annotation data are tuples containing the ID of the annotated description, the LOC of the DF at that moment (from the VR engine log), and a class label: either Positive or Negative. Because we expect some latency between when the DG judges that a felicity condition is met and when he begins to speak, rather than using spatial context features that co-occur with the onset of each description, we averaged the values over a 0.3 second window centered at the onset of the expression.

Negative contexts are difficult to identify since they often do not manifest linguistically: the DG may say nothing and allow the user to continue moving along his current vector, or he may issue a movement command. A minimal criterion for producing an expression that can achieve joint spatial reference is that the addressee must have perceptual accessibility to the item. Therefore, negative training examples for this experiment were selected from the time-

periods that elapsed between the follower achieving perceptual access to the object (coming into the same room with it but not necessarily looking at it), but before the Locating description was spoken. In these negative examples, we consider the basic felicity conditions for producing a descriptive reference to the object to be met, yet the DG did not produce a description. The dataset of 932 training examples was balanced to contain 50% positive and 50% negative examples.

3.2 Decision Tree Performance

This evaluation is based on our algorithm's ability to reproduce the linguistic behavior of our human subjects, which may not be ideal behavior.

The Weka¹ toolkit was used to build a decision tree classifier (Witten and Frank, 2005). Figure 4 shows the resulting tree. 20% of the examples were held out as test items, and 80% were used for training with 10 fold cross validation. Based on training results, the tree was pruned to a minimum of 30 instances per leaf. The final tree correctly classified 86% of the test data.

The number of positive and negative examples was balanced, so the first baseline is 50%. To incorporate a more elaborate baseline, we consider that a description will be made only if the referent is visible to the DF. Marking all cases where the referent was visible as *describe-id* and all the other examples as *delay* gives a higher baseline of 70%, still 16% lower than the result of our tree.²

Previous findings in spatial cognition consider angle, distance and shape as the key factors establishing spatial relationships (Gapp, 1995), the angle deviation being the most important feature for projective spatial relationship. Our algorithm also selects **Angle** and **Distance** as informative features. **VisDistracts** is selected as the most important feature by the tree, suggesting that having a large number of objects to contrast makes the description harder, which is in sync with human intuition. We note that Visible is not selected, but that might be due to the fact that it reduces to $\text{Angle} > 50^\circ$. In terms of the referring expression generation algorithm described by (Reiter and Dale, 1992), in which the description which eliminates the most distractors is selected, our

¹<http://www.cs.waikato.ac.nz/ml/weka/>

²not all positive examples were visible

results suggest that the human subjects chose to reduce the size of the distractor set before producing a description, presumably in order to reduce the computational load required to calculate the optimal description.

```

VisDistracts <= 3
| Angle <= 33
| | Distance <=154: describe-id (308/27)
| | Distance > 154: delay (60/20)
| Angle > 33
| | Distance <= 90
| | | Angle <=83:describe-id(79/20)
| | | Angle > 83: delay (53/9)
| | Distance >90: delay(158/16)
VisDistracts > 3: delay (114/1)

```

Figure 4: The decision tree obtained.

Class	Precision	Recall	F-measure
describe-id	0.822	0.925	0.871
delay	0.914	0.8	0.853

Table 1: Detailed Performance

The exact values of features shown in our decision tree are specific to our environment. However, the features themselves are domain-independent and are relevant for any spatial direction-giving task, and their relative influence over the final decision may transfer to a new domain. To incorporate our findings in a system, we will monitor the user’s context and plan a description only when our tree predicts it.

4 Conclusions and Future Work

We describe an experiment in content planning for spoken dialog agents that provide navigation instructions. Navigation requires the system and the user to achieve joint reference to objects in the environment. To accomplish this goal human direction-givers judge whether their partner is in an appropriate spatial configuration to comprehend a reference spoken to an object in the scene. If not, one strategy for accomplishing the communicative goal is to steer their partner into a position from which the object is easier to describe.

The algorithm we developed in this study, which takes into account spatial context features replicates our human subject’s decision to produce a description with 86%, compared to a 70% baseline based on the visibility of the object. Although the spatial details will vary for other spoken dialog domains, the process developed in this study for producing description dialog moves only at the appropriate times

should be relevant for spoken dialog agents operating in other navigation domains.

Building dialog agents for situated tasks provides a wealth of opportunity to study the interaction between context and linguistic behavior. In the future, the generation procedure for our interactive agent will be further developed in areas such as spatial descriptions and surface realization. We also plan to investigate whether different object types in the domain require differential processing, as prior work on spatial semantics would suggest.

5 Acknowledgements

We would like to thank the OSU CSE department for funding this work, our participants in the study and to M. White and our reviewers for useful comments on the paper. We also thank Brad Mellen for building the virtual world.

References

- D. J. Bryant, B. Tversky, and N. Franklin. 1992. Internal and external spatial frameworks representing described scenes. *Journal of Memory and Language*, 31:74–98.
- D. K. Byron. 2005. The OSU Quake 2004 corpus of two-party situated problem-solving dialogs. Technical Report OSU-CISRC-805-TR57, The Ohio State University Computer Science and Engineering Department, Sept., 2005.
- J. Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.
- R. Dale, S. Geldof, and J. Prost. 2003. CORAL: Using natural language generation for navigational assistance. In M. Oudshoorn, editor, *Proceedings of the 26th Australasian Computer Science Conference*, Adelaide, Australia.
- K. Gapp. 1995. Angle, distance, shape, and their relationship to projective relations. Technical Report 115, Universitat des Saarlandes.
- M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. 2002. MATCH: An architecture for multimodal dialogue systems. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL ’02)*, pages 376–383.
- M. Kipp. 2004. *Gesture Generation by Imitation - From Human Behavior to Computer Character Animation*. Dissertation.com.
- R. Moratz and T. Tenbrink. 2003. Instruction modes for joint spatial reference between naive users and a mobile robot. In *Proc. RIISP 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, Special Session on New Methods in Human Robot Interaction*.
- C. Muller. 2002. Multimodal dialog in a pedestrian navigation system. In *Proceedings of ISCA Tutorial and Research Workshop on Multi-Modal Dialogue in Mobile Environments*.
- E. Reiter and R. Dale. 1992. A fast algorithm for the generation of referring expressions. *COLING*.
- B. Tversky and P. U. Lee. 1999. Pictorial and verbal tools for conveying routes. Stade, Germany.
- I. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, San Francisco.