Direction giving: an attempt to increase user engagement

Bob Duncan and Kees van Deemter
Computing Science department, University of Aberdeen
(email: r.duncan.07@aberdeen.ac.uk, k.vdeemter@abdn.ac.uk)

Abstract

These notes describe a contribution to the 2011 GIVE Challenge from the University of Aberdeen. Our contribution focuses on an attempt to increase the extent to which participants felt engaged in the direction giving/following game on which the GIVE challenge focuses.

1 Introduction

These notes outline the first author's (undergraduate) final-year Computing Science project. Its main aim was to give the authors a hands-on understanding of the GIVE framework, and to see whether this framework should play a role in their future research on the generation of referring expressions (GRE). Our motivation was that previous assessments of GRE algorithms (Jordan and Walker 2005, Viethen and Dale 2007, Gatt and Belz 2010, Van Deemter et al. 2011) have typically focused on simplified experimental settings, where the domain is very small, and where the location of the hearer and speaker is not taken into account as a factor that influences the salience of the different domain objects. GIVE offers the possibility of doing away with these limitations in a rich, semi life-like environment, hence our interest.

The GIVE challenges place participants in a virtual world where they are going on a treasure hunt. To find the treasure, participants need to navigate through a building and push a series of buttons. GIVE asks for the submission of algorithms that help participants perform their treasure hunt. They should help them navigate through the building, and push the right buttons (while carefully avoiding others, which may set off alarms). An informal exploration of the systems submitted to the previous (2009) GIVE challenge suggested to us that there were three main areas in which there was substantial room for improvement of the algorithms submitted then: (1) user engagement, (2) special gadgets that might assist the user in his/her quest, and (3) the quality of the referring expressions generated. We elaborate briefly on each of these factors.

2 The Aberdeen system

2.1 User engagement

Subjective comments from participants to GIVE-2009 (see Koller et al. 2010) suggest that the algorithms submitted at the time were not well able to ``engage'' participants in the task, which may have felt more like a chore to them than like an enjoyable game. It seemed plausible that if user engagement could be improved, this would not only be a good thing in its own right, but that it might also lead to improved results on objective task performance metrics such as task completion rates (cf. Lester et al. 1997). In view of these observations, we attempted to increase users' engagement in the game by adding a "James Bond" theme to the utterances generated by the system. At the start of the game, for example, the system says: *"Hello, James Bond, Secret Agent 007, welcome to the GIVE World! Your mission is to get a trophy full of diamonds from a safe. To do this, you must turn off alarms, uncover the safe, and crack the safe combination. Now pay attention 007. I need to tell you three very important things: One, you need to get really close to a button before you press it! Two, if there is no message, go to the middle of the room to re-activate the scanner! Three, don't stand on the red tiles, 007. They are all alarmed!"*

Proceedings of the 13th European Workshop on Natural Language Generation (ENLG), pages 280–283,
Nancy, France, September 2011.

2.2 Gadgets

GIVE offers an electronic "world" that differs from real life. It seemed reasonable to us to make use of this fact by allowing the user to do things that might be impossible in real life. In particular, we decided to offer users the use of a gadget that we called ATAC (Automatic Target Acquisition Control). When activated, ATAC detects the correct target (for example, the button that needs to be pressed at a given moment in time) then checks whether it is "in view" (i.e., nearby). If it is, the system says "target acquired", otherwise it says that the target is not there. ATAC was expected to be particularly useful in preventing participants from pushing alarmed buttons.

2.3 Referring Expressions

A quick survey of the systems submitted to GIVE 2009 suggested to us that generation of referring expressions was generally a weak point. A good example is Denis (2009), which appears to rely on a strategy whereby the system indicates an underspecified referring expression (e.g., *"(push) a red button"*); if the user pushes the wrong button, the system proceeds to say that the wrong button was pushed, and another one needs to be attempted. While it is interesting to have a referential strategy that allows a degree of collaboration between speaker and hearer (cf. Heeman and Hirst 1995), this particular strategy seems error prone (particularly given the existence of alarmed buttons), and problematic in the presence of a large domain. (What if there are 10 red buttons, for example?)

Our initial plan was to use the algorithm of van Deemter (2006), originally designed to generate vague descriptions such as "The tall man". In a configuration of buttons on a wall, for example, this algorithm is able to identify any single button, by generating a sequence of gradable properties. Imagine a sequence of three buttons, for example, numbered 1,2,3 from left to right. Button 2 may be identified by the sequence "*Take the leftmost two buttons", "(From these) take the rightmost button"*. The problem, however, lies in Linguistic Realisation: a direct rendering of the sequence would give rise to a highly complex description, whereas an optimal rendering would simply say "The button in the middle". Programming this nontrivial Linguistic Realisation step proved too difficult a task within a final-year project that was full of other challenges. Moreover, the ATAC gadget (section 2.2) offers the user an additional technique, which might make complex referring expressions unnecessary in most situations. For these reasons, we decided to explore an alternative approach, which distinguishes a number of different referential situations, each of which is addressed by a largely separate procedure (though code was shared between these procedures as much as possible). Essentially, we used a large battery of small algorithms; an appropriate algorithm was chosen depending on the situation. This inelegant but flexible "engineering" approach made it easy for us to address a number of special situations which are often disregarded (e.g., the situation where the domain does not contain any distractors). It works by distinguishing a series of increasingly complex referential situations (programmed as CASE statements), starting with the simplest situations that a GIVE participant can encounter, and ending with the most complex ones. (In the list of cases, each case assumes that previous cases do not apply.)

CASE 1: There is only one button in the room, and this button is the target. System (example): *"There is a single blue button in this room. Push it, James!"*

CASE 2: The target button is the only one in its target region. System: *"There is a single button on the left wall. Push it."*

CASE 3: The target button has a colour that is unique in its target region. System: *"There is a row of four buttons on your right. Press the red button."*

CASE 4: There exists in the target region just one (horizontal or vertical) sequence of buttons, and the target button is one of these buttons. System: *"There is a horizontal sequence of buttons on your left. Push the rightmost button in this sequence."*
…

CASE n:
System: *"Use the ATAC scanner, James!"*

3 Evaluation of the Aberdeen system

The "objective" performance of our system, in terms of task completion percentages, times and words was largely unremarkable. In fact, our "James Bond" theme made our system more verbose than most, and the navigation aspect of our system drew a number of negative comments from participants, particularly regarding the timing of the system's messages (*"The system reacted very slowly on my progress. The commands were designed for really slow steps while I'm used to 'walk' quickly", "The message 'go through the doorway' was always too late", "The speed of the commands were a little bit too late."*) For details concerning objective performance, we refer to the organisers' figures. Here, we will attempt to assess to what extent the three innovations discussed in section 2 were successful. In each case, we start summarizing relevant parts of the questionnaire, followed by a summary of comments.

User engagement.
Questionnaire: The subjective questions did not address the extent to which a system managed to "engage" the user in the direction-giving game. Consequently, they did not shed light on our claim, neither confirming nor disconfirming it.
Comments: *"The fact that the system tells us that we are a secret agent, that's cool", "The salutation with 007 was very funny", "Altogether an acceptable game", "It was a fun game to play while it lasted."*

Gadgets.
Questionnaire: The subjective questions did not address this issue.
Comments: *"Saying 'target not here' or 'target in front of you' helped in letting me know if I'd reached the right place".*

Referring Expressions.
Questionnaire: The analysis of subjects' responses to the statement in the questionnaire that said "I could easily identify the buttons the system described to me" appears to confirm that the referring expressions produced by our system were clear. The results in this area were not statistically significant, however, so need to be treated with caution.
Comments: *"I'm impressed by the overall quality of the instructions I received. As an AI researcher I'm interested in such endeavors and will follow the progress in the near future", "The system worked better when I was near the correct buttons and it gave explicit instructions about which button to press", "It was quite good in describing which button was to be pressed", "The descriptions of which buttons to press were generally clear", "The descriptions of which buttons to push was quite clear", "The description of the buttons was most of the times unambiguous", "Good instructions", "Liked description of colors of buttons, numbers of buttons", "It's very good describing buttons positions, and has good relative references", "The button finding instructions were very easy to follow", "The identification of the buttons one must press is done almost impeccably."*

As it happens, these aspects of the system appeared to give rise to almost exclusively positive comments. Perhaps these positive comments need to be taken with a pinch of salt, given that they did not translate into better "objective" performance. (Compare Dehn and Van Mulken 2000 for a discussion of a similar asymmetry between subjective experience and objective task performance, in the area of Embodied Conversational Agents.)

4 Conclusion and general notes on the GIVE challenge

Mastering the GIVE software proved a major challenge for us, especially after the system was installed in the network, when a variety of new issues arose, relating to the use of ports, proxies and permissions. Taking part in GIVE became a very "technical" affair, with issues of Natural Language Generation and HCI taking a definite backseat.

We expect that researchers who want to use the GIVE framework itself (rather than participate in the GIVE challenge) are unlikely to experience these problems, however, because their programs will not need to be installed into the network. In regard of our plans to use the GIVE setting for our own future experiments, this is an encouraging conclusion.

In our initial exploration, we underestimated the

problems thrown up by navigation. Users can easily feel disoriented when they end up in an area where they should not be. Equally, if the user moves faster than the system can keep up with (in terms of producing the next instruction) then instructions can arrive too late to be of relevance, which can further disorient the user. Tackling these issues required more attention than we had anticipated. Having said this, it appears that those aspects of the system on which we decided to focus (user engagement, the ATAC gadget, and referring expressions) were fairly successful

Acknowledgments

We are grateful to Roman Kutlak and Margaret Mitchell for their assistance troubleshooting software issues, and to Chris Mellish for general advice.

References

Dehn and Van Mulken 2000. D.M. Dehn, S. van Mulken. The impact of Interface Agents: a Review of Empirical Research. *Journal of Human-Computer Studies* 52(1), p.1-22.

Denis, Alexandre. 2010. Generating referring expressions with reference domain theory. In Proceedings of the 6th International Natural Language Generation Conference
(INLG), Trim, Ireland.

Gatt, Albert and Anja Belz. 2010. Introducing shared task evaluation to NLG: The TUNA shared task evaluation challenges. In Emiel Krahmer and Mariët Theune (Eds.) Empirical Methods in Natural Language Generation. Springer Verlag, Berlin, pages 264–293.

Goudbeek, Martijn and Emiel Krahmer. 2010. Preferences versus adaptation during referring expression generation. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), pages 55–59, Uppsala, Sweden.

Gupta, Surabhi and Amanda Stent. 2005. Automatic evaluation of referring expression generation using corpora. In Proceedings of the 1st Workshop on Using
Copora in Natural Language Generation (UCNLG), pages 1–6, Brighton, UK.

Heeman, Peter A. and Graeme Hirst. 1995. Collaborating on referring expressions. *Computational Linguistics*, 21(3):351–382.

Jordan, PamelaW. and Marilyn Walker. 2005. Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research,* 24:157–194.

Koller, Alexander, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. The first challenge on generating instructions in virtual environments. In Emiel Krahmer and Mariët Theune, editors, Empirical Methods in Natural Language Generation. Springer Verlag, Berlin, pages 328–352.

Lester, J.C., S.A.Converse, S.E.Kahler, S.T. Barlow, B.A. Stone, R.S.Bhoga,. The Persona Effect.: Affective Impact of Animated Pedagogical Agents, in: Proc. CHI Conference, Atlanta, Georgia.

van Deemter, Kees. 2006. Generating Referring Expressions that involve gradable properties. *Computational Linguistics*, 32(2):195–222.

van Deemter, Kees, Albert Gatt, Ielka van der Sluis, and Richard Power (in press). Generation of Referring Expressions: Assessing the Incremental Algorithm. *Cognitive Science*. To appear Winter 2011-2012.

Viethen, Jette and Robert Dale. 2007. Evaluation in natural language generation: Lessons from referring expression
generation. *Traitement Automatique des Langues*, 48:141 –160.