The Potsdam NLG systems at the GIVE-2.5 Challenge

Konstantina Garoufi and **Alexander Koller**
Area of Excellence "Cognitive Sciences"
University of Potsdam, Germany
{garoufi, akoller}@uni-potsdam.de

Abstract

We present the Potsdam natural language generation systems P1 and P2 of the GIVE-2.5 Challenge. The systems implement two different referring expression generation models from Garoufi and Koller (2011) while behaving identically in all other respects. In particular, P1 combines symbolic and corpus-based methods for the generation of successful referring expressions, while P2 is based on a purely symbolic model which serves as a qualified baseline for comparison. We describe how the systems operated in the challenge and discuss the results, which indicate that P1 outperforms P2 in terms of several measures of referring expression success.

1 Introduction

The Challenge on Generating Instructions in Virtual Environments (GIVE; Koller et al. (2010)) is an evaluation effort for natural language generation (NLG) systems, which focuses on real-time generation of situated language. In this shared task, the role of the NLG system is to guide a human instruction follower (IF) through a 3D virtual world with the goal of completing a treasure-hunting task. As an internet-based evaluation, GIVE has been successful in attracting both a large number of volunteers for the IF role and a high level of interest from the research community.

In this paper, we report on our participation in the third installment of GIVE (GIVE-2.5; Striegnitz et al. (2011)). Although most of the work on the generation of referring expressions (REs) to date has focused either on logical properties of REs, such as uniqueness and minimality, or on their degree of similarity to human-produced expressions (see Krahmer and van Deemter (To appear) for a comprehensive survey), we believe that it would be desirable to optimize a system directly for usefulness. We therefore approach the RE generation task with a model that aims at computing the unique RE which is fastest for the hearer to resolve (Garoufi and Koller, 2011). The purpose of the Potsdam NLG systems P1 and P2 at the challenge was to assess with a task-based evaluation to what extent the model actually manages to do so.

While we cannot present the RE generation modules in detail here (see Garoufi and Koller (2011) for that), note that P1 implements the hybrid model mSCRISP of Garoufi and Koller, which extends the planning-based approach to sentence generation (Koller and Stone, 2007) with a statistical model of RE success. This model was learnt from a corpus of human instruction giving sessions in the GIVE domain (Gargett et al., 2010), in which every RE was annotated with a measure of how easy it has been for the hearer to resolve. System P1 is therefore designed to optimize the REs it generates for understandability. On the other hand, system P2 is an implementation of the baseline model EqualCosts of Garoufi and Koller. This is a purely symbolic model that always computes a correct and unique RE, but does so without any empirical guidance about expected understandability. System P2 behaves in the exact same way as P1 in all respects, with the exception of the RE generation module. It therefore serves as a qualified baseline against which we can

Proceedings of the 13th European Workshop on Natural Language Generation (ENLG), pages 307–311,
Nancy, France, September 2011. ©2011 Association for Computational Linguistics

compare the performance of the mSCRISP model.

Plan of the paper. We describe the two systems P1 and P2 in Section 2. As the RE generation modules have been presented in full detail in Garoufi and Koller (2011), we mostly focus on the other aspects of the systems' behavior here. We then comment on the evaluation results in Section 3 and conclude in Section 4.

2 The systems P1 and P2

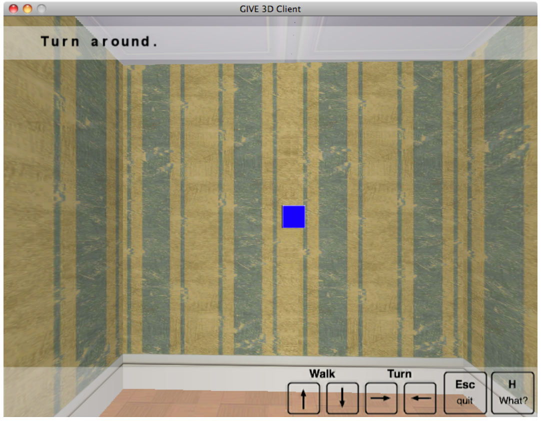
The two systems operate on the same codebase, differing only in their RE generation modules. In particular, they follow identical strategies for determining their communicative goals, switching between navigation and reference, as well as issuing warnings and other feedback.

2.1 Determining the communicative goals

The GIVE framework provides an NLG system with a plan of what the IF must do in order to complete the task by picking up a trophy. This plan is a symbolic sequence of mixed moves and object manipulation actions such as move(reg_1, reg_2), manipulate$(b_1, \mathit{off}, on, reg_2)$, take–t$_1(reg_3)$. Our systems parse the plan in order to identify objects of interest and determine the nature of the communicative goals related to these: If a move action which involves going through a doorway from one room to another is encountered in the plan, then that doorway is registered as a target with the corresponding communicative goal that the IF should go through it. If, on the other hand, a manipulate or take action is encountered, then the patient of this action is registered as a target (be it a button to push or a trophy to take), while, accordingly, the manipulation of that target becomes a communicative goal for the systems to pursue.

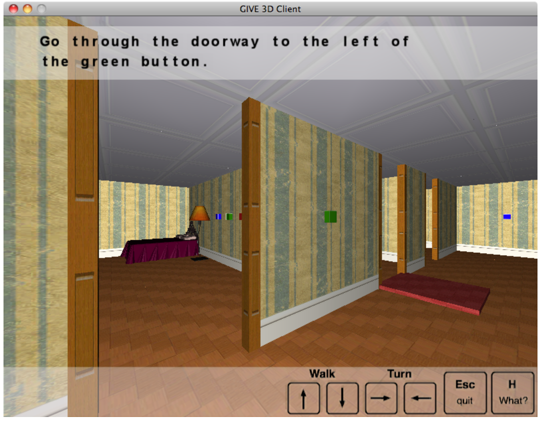
2.2 Navigation and reference

Once the next target and the communicative goal have been determined, the systems go on to check whether a certain condition for reference is met; in particular, whether the target is currently in the IF's field of view. This precondition reflects empirical observations that human instruction givers typically manipulate the non-linguistic context of scenes in convenient ways (e.g. by making the referent visually salient) before referring to objects in these scenes (Stoia et al., 2006; Schütte et al., 2010). If the precondition is not fulfilled, then the systems resort to low-level navigation instructions such as "Turn left" or "Go straight" in order to change the IF's location to one that allows them to see the target (Figure 1). Because doorways are also perceived as targets, it is guaranteed that the next target is always located in the same room as the IF. As a result, this process usually involves no more than a few turns.



Turn around.

Figure 1: Example of a navigation instruction aiming at making the next target visible.



Go through the doorway to the left of the green button.

Figure 2: Example of a navigation instruction urging the IF to go through a doorway that they already see.

Once the target has become visible, the systems switch to referring expression generation mode so as to issue an instruction that describes the target and satisfies the communicative goal. Note that although the evaluation is concerned with REs to button targets only, we apply the same RE generation models to the description of all objects, including doorways and the trophy. Figure 2 shows an example of a navigation instruction that urges the IF to go through a visible doorway, while Figure 3 presents an example of an RE for a button target issued by system P1. In this scene, system P2 would generate the different RE “the right one to the right of the green button”. The systems issue all these kinds of instructions at regular intervals repeatedly, until they detect that the IF has reacted. This is to make sure that the IF knows at all times what they are expected to do.

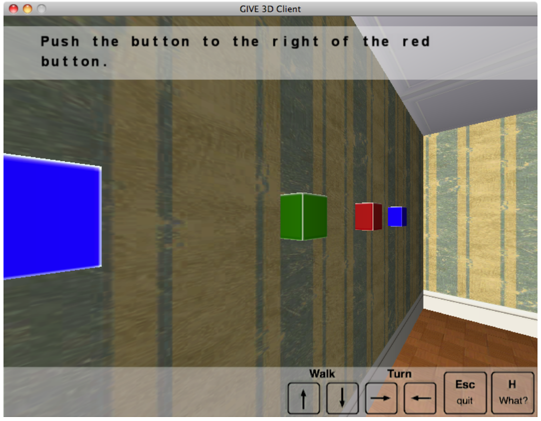
2.3 Execution monitoring

In real-time instruction giving it is crucial for a system to be able to monitor whether the IF actually executes the given instructions, assess how well they progress on the task, and finally react to such observations with appropriate feedback. Our systems issue three main types of such feedback:

- **Positive feedback.** The IF receives an affirmation (e.g. “Good job!”, “Excellent!”) as soon as they accommodate the given communicative goal by executing the associated action. These situations are important because apart from moving the task forward they establish that a system’s RE has been resolved by the IF correctly.

- **Negative feedback.** Conversely, if the IF performs a different action than the one expected, e.g. by pushing the wrong button or going into the wrong room, they are immediately told so (Figure 4). This serves not only as feedback for the IF but also as an opportunity for the systems to reevaluate the situation and make the necessary computations for figuring out which communicative goal should come next.

- **Warnings.** Finally, certain regions in the GIVE worlds are equipped with alarms so that stepping on them would cause the IF to lose the game. If the systems detect that the IF has approached an activated alarm closely enough that this outcome becomes likely, they interrupt all their other functions and issue a brief warning about the danger (e.g., “Beware of the alarm on the floor!”).


GIVE 3D Client

Push the button to the right of the red button.

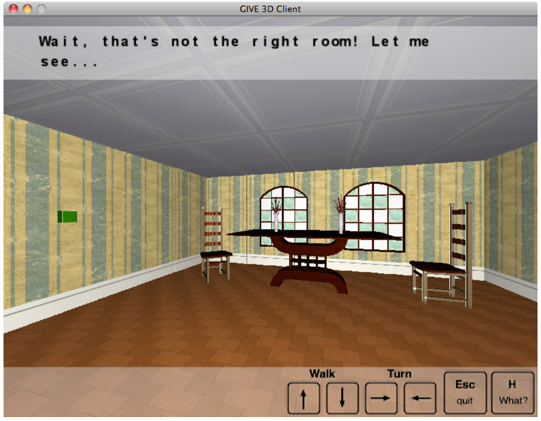
Figure 3: Example of an expression referring to a button target, as generated by P1.

2.4 Example instruction-giving session

Example (1) below presents a simplified excerpt from an interaction between system P1 and an IF, in which several of the instruction types listed above can be found.

(1) P1: *Turn left.*
IF: (turns left until the target becomes visible)
P1: *Push the yellow button.*
IF: (starts moving towards the button)
P1: *Push the button.*
IF: (pushes the button)
P1: *Good!*
P1: *Now turn right.*
IF: (turns right until the target becomes visible)
P1: *Go through the doorway.*
IF: (goes through the doorway)
P1: *Excellent!*
P1: *Turn right.*
IF: (starts turning right)
P1: *Go straight.*
IF: (starts moving straight ahead)
P1: *Don’t step on the alarm!*
P1: *Go straight.*
IF: (continues moving ahead)
P1: *Push the green one in front of you.*

One meaningful detail is that, as lines 3–5 reveal, the REs that the system generates for a given target may change as the context of the scene changes. This particularly interesting aspect of the interaction follows from the fact that the system generates its REs newly for every new context, and thus decides newly which



Wait, that's not the right room! Let me see...

Figure 4: Example of execution monitoring and negative feedback.

attributes to include in it and which not. Since the attribute selection process of P1 relies on the context features of the scene in a much more substantial way than that of P2, which simply uses the visual context in order to ensure that the RE is distinguishing in the domain, this phenomenon is observed in P1 more frequently than in P2. Indeed, P1 may change its decision of which attributes to include in an RE not because, say, a potential distractor has come into sight, but just because e.g. the IF has moved closer to the target, or even because the system has already attempted to refer to it in a particular way several times before without success.

3 Results

Although none of the objective and subjective evaluation measures of the challenge establish any significant differences between the two systems based on the current snapshot of the results, P1 does achieve better scores than P2 on most measures of RE success.

3.1 RE success

Areas in which P1 outperforms P2 include the objective measures of task success, number of actions executed by the IF (indicating incorrect resolution of REs), and game duration. But also in terms of subjective measures, as extracted by the IFs' responses to a post-task questionnaire, P1 scores higher than P2 for the most part: It is perceived as generating better instructions overall ("Overall, the system gave me good instructions"), better REs to buttons ("I could easily identify the buttons the system described to me"), and clearer, more trustworthy instructions ("I had to re-read instructions to understand what I had to do", "I felt I could trust the system's instructions"; see Striegnitz et al. (2011) for details).

More importantly, we compared the systems with respect to RE resolution success and successfulness, which is the exact measure of RE understandability that P1 was optimized for. This comparison does establish a significant difference between the two, indicating that P1 generates REs that are faster resolvable by the IFs after effects of RE rephrasing as described in Subsection 2.4 have been factored out (see Garoufi and Koller (2011) for details).

3.2 Error analysis

Finally, looking into possible causes of failure for the systems' REs, we find that the most apparent problem involves generating expressions which, though not semantically invalid, are of disputable linguistic acceptability. Typical instances of such REs are "the button to the left of the right button", "the button below the upper button" and variants of these. These cases arise due to the fact that we did not constraint the systems' grammar so as to disallow such constructions, while the systems often chose attributes for which this particular type of realization was possible.

It turns out that P2 was more prone to this type of REs than P1, which at first glance seems like a probable reason for the lower RE success rates of the system. However, examining the portion of REs of each system that did not fall into this category, we found that P1 still generated significantly more successful REs after factoring out the effects of rephrasing. It would be interesting for future work to compare the systems' REs in a more controlled way, so that their attribute selection and realization aspects can be evaluated in separation.

4 Conclusion

The systems P1 and P2 at the GIVE-2.5 Challenge implemented a novel model of RE generation and a qualified baseline from Garoufi and Koller (2011), respectively. Participating in the challenge allowed us to conduct a task-based evaluation of the model, collect data for both objective and subjective measures, and compare it against the baseline. The results indicate that the model outperforms the baseline with respect to the measure of RE understandability that it was optimized for.

Acknowledgments

Thanks to Kristina Striegnitz for clarifying important aspects of the evaluation methodology, and to all 148 internet users who went on a virtual treasure hunt to make this evaluation possible.

References

Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The GIVE-2 Corpus of Giving Instructions in Virtual Environments. In *Proceedings of the 7th Conference on International Language Resources and Evaluation*, Valletta, Malta.

Konstantina Garoufi and Alexander Koller. 2011. Combining symbolic and corpus-based approaches for the generation of successful referring expressions. In *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France.

Alexander Koller and Matthew Stone. 2007. Sentence generation as planning. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic.

Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. The First Challenge on Generating Instructions in Virtual Environments. In M. Theune and E. Krahmer, editors, *Empirical Methods in Natural Language Generation*, volume 5790 of *LNCS*, pages 337–361.

Emiel Krahmer and Kees van Deemter. To appear. Computational generation of referring expressions: A survey. *Computational Liguistics*.

Niels Schütte, John Kelleher, and Brian Mac Namee. 2010. Visual salience and reference resolution in situated dialogues: A corpus-based evaluation. In *Proceedings of the AAAI 2010 Fall Symposium on Dialog with Robots*, Arlington, VA.

Laura Stoia, Donna K. Byron, Darla M. Shockley, and Eric Fosler-Lussier. 2006. Sentence planning for realtime navigational instructions. In *Proceedings of the Human Language Technology Conference of the NAACL*, New York City, NY.

Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariet Theune. 2011. Report on the Second Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2.5). In *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France.