Two approaches to GIVE: dynamic level adaptation versus playfulness

Roan Boer Rookhuiszen, Michel Obbink, Mariët Theune
Human Media Interaction
University of Twente
Enschede, The Netherlands
{a.r.boerrookhuiszen, m.obbink}@student.utwente.nl
m.theune@ewi.utwente.nl

Abstract

In this paper we introduce two NLG systems that we developed for the GIVE challenge. One system focuses on generating optimally helpful instructions while the other focuses on entertainment. We used the data gathered in the Challenge to compare the efficiency and entertainment value of both systems. We were unable to prove that one system was more entertaining than the other. Furthermore, we discuss remarkable results and give some suggestions for future editions of the GIVE Challenge.

1 Introduction

In this paper, we present the NLG systems we developed for the Challenge on Giving Instructions in Virtual Environments (GIVE), an NLG evaluation challenge to generate instructions for users in a game-like 3D environment. In the challenge, users were asked to perform a task in a game-like 3D virtual environment. To 'win' the game, they had to follow the instructions produced by an NLG system. The objective of the GIVE-game for the player is to find a trophy without triggering an alarm. The trophy is hidden in a safe behind a picture on one of the walls. The safe can only be opened by pressing several buttons in the right order. The evaluation in GIVE is done based on performance data and subjective ratings gathered with a user questionnaire. A website[1] was set up with a short instruction and the game.

We participated in the GIVE Challenge with two NLG systems: one system that was focused on generating maximally helpful instructions (the Twente system) and one that was intended to be more game-like and thus entertaining (The Warm/Cold game).

Although the GIVE Challenge was presented as a game to its users, who were invited to 'play a game', the evaluation criteria used in the Challenge still focused on effectiveness and efficiency of the generated instructions. In other words, the NLG systems were evaluated as if used in a serious application rather than a game. Nevertheless, in this paper we will try to use the GIVE evaluation data for comparing our own systems in terms of not only efficiency, but also entertainment.

In the following sections, we first describe the two systems we developed for the GIVE Challenge (Section 2), followed by a brief discussion of the most remarkable results from the GIVE evaluation (Section 3). Then we describe how we used the evaluation data to determine if the Warm/Cold system was indeed more entertaining than the Twente system (Section 4). We end with some suggestions for future editions of the GIVE Challenge (Section 5).

2 Our NLG systems

For the Challenge, we designed two NLG systems, each with a different goal:

1. The Twente system, focusing on efficiency
2. The Warm/Cold system, focusing on entertainment

The first system, the Twente system, is purely task-oriented and tries to guide the user through the game as efficiently as possible. The Warm/Cold system on the other hand tries to make the game more entertaining for the user even if a consequence is a decrease of the efficiency. Below we describe both systems.

2.1 The Twente system

The organization of the GIVE Challenge provided all participating teams with an example implementation in Java of an NLG system. This system was

[1] http://www.give-challenge.org/

very basic and instructed the user to carry out only one action at a time (e.g., take one step forward). This was easy to understand, especially for new users; however it was very annoying for more experienced users. In our first attempt at implementing our own NLG system, all instructions to reach a button were combined into one sentence. More experienced users did perform better with this system than with the example implementation (they used less time, and found it more clear), but inexperienced users made more errors with the more complex sentences. Because of this difference between new and more experienced users we decided to design an adaptive framework with three different levels. The first level generates very basic instructions, explicitly mentioning every step of the plan. The higher levels generate more abstract, global instructions that are expressed using more complex sentences. Some example sentences generated by the different levels:

- Level 1: Only one instruction at a time: "Walk forward 3 steps", "Press the blue button", "Turn right."
- Level 2: A combination of a walk instruction and another action instruction: "Walk forward 3 steps then press the button."
- Level 3: Also a combination, but only referring to objects when the user could see them: "Turn right and walk forward", followed by "Press the blue button." In this level we thus do not give the exact route to the next button to be pushed, but try to encourage users to walk to it on their own once they have it in view.

All levels are generated using the same general framework. Certain actions are the same for all levels: interpreting events, the generation of referring expressions ("The blue button"), the check whether users are performing the correct actions and the check whether the level should be changed. The differences between the levels can be found in the wording of the instructions and their timing, which is different for each level: at the first level a new, updated instruction is given after each step made by the user, but at the second level it is given as soon as the player has fulfilled the previous instruction. At the third level, the system first gives an instruction to indicate the direction in which the user has to walk. As soon as the next goal (i.e., button to be pushed) is visible a new instruction referring to that goal is generated. A new instruction is also generated whenever the user goes into the wrong direction.

The sentences generated by the different levels are fairly similar. They are generated by using small templates that are different between levels, but all templates just need the referring expressions and the number of steps to be filled in. The system makes use of simple referring expressions to refer to buttons. When the intended referent is the only visible button with a particular color, the color is used to distinguish this button from the others ("Press the green button"). If another visible button happens to have the same color, the relative position is used ("The second blue button from the right").

For each level we have developed a function to determine whether a new instruction should be presented to the user and a function that generates the instruction. The first function is called every second, and directly after the user takes a step or performs an action. When the first function decides that a new instruction sentence has to be generated, the second function is executed. The input and output parameters for those functions are the same for each level, only their specific implementation is different. Because of this similarity the levels can easily be switched, and new levels can easily be added to the framework.

We started with implementing the first level. Based on this first implementation the other levels were created. After the first implementation we asked a few users to play each level of the game separately (so no automatic switching of levels). During these tests, we received suggestions for several small adaptations. For example, in our first version of level 2 a sentence consisted of an action followed by a move instruction. For example:"Turn right then walk 3 steps". People found it more natural and easier to understand when the order was changed to having a move followed by an action: "Walk 3 steps then turn right."

Our framework is *adaptive*; the NLG system will try to fit the level to the user's needs. It is expected that novice users learn while they are playing the game. The system is able to detect the level of experience of the user and automatically change the level during the game. When the game starts the used level is 2. Every second, the system checks the number of actions the user performed

Table 1: List of thresholds used for automatic level switching, in terms of the number of actions performed in the last 5 seconds.

Level change	Threshold
1 up to 2	$>$ 5 actions
2 up to 3	$>$ 8 actions
2 down to 1	$<$ 2 actions
3 down to 2	$<$ 3 actions

in the last 5 seconds. Whenever this number exceeds a certain threshold the user will probably perform better on a higher level, so the level is switched upward. On the other hand the level is switched down as soon as the number of actions is low or the user presses the Help button.

We have used different values as a threshold to determine if the level should be higher or lower. The thresholds between Levels 1 and 2, and between Levels 2 and 3 are also different. In the higher levels the user is performing faster, resulting in more actions in 5 seconds. However, in the higher levels the user has to read more text, resulting in the user performing no actions for some time while reading the instructions. The level should not switch down immediately when this happens and therefore the thresholds are set relatively high.

We have determined the thresholds by letting players play the game in a fixed level for several times. We clearly saw a learning effect: players were much faster the second time they played the game. The list of used thresholds can be found in Table 1. The level is increased by one as soon as the number of actions in the last 5 seconds is more than the threshold and decreased when the number of actions in the last 5 seconds is less than the given threshold.

2.2 The Warm/Cold system

To make the task more interesting for the users, we created a more game-like situation. This implementation tries to simulate a warm/cold game. Instead of telling the user exactly what to do, the only instructions given are “warmer” and “colder” to tell the user if he comes closer to the next button to be pushed, “turn” to indicate that the user only has to turn to see that button and of course the instruction to push it.

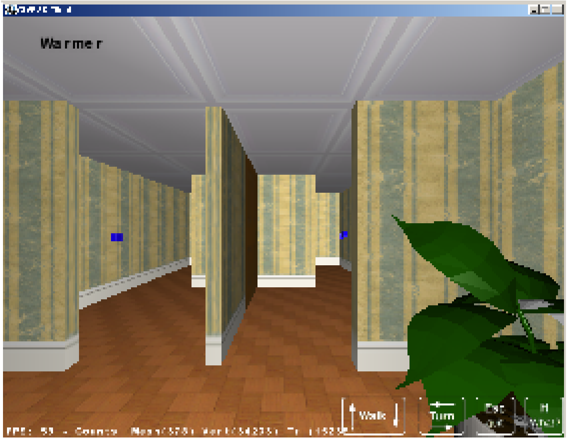

Figure 1: The Warm/Cold system tells the user he is getting “Warmer” (i.e., closer to the next target button).

Thus, most of the utterances generated by the system are ‘hints’ rather than straightforward action instructions. Before the user gets his first hint, he has to walk around in any direction. Then he can use the “warmer” / “colder” information to decide in which direction to go next. The information given by the system is ambiguous; it does not tell the user where to go but leaves the navigation choices open. This is illustrated in Figure 1: the user is warned that he is getting closer (“warmer”) to the button to be pushed, but he still has to decide for himself whether to go left or right.

Moreover, to find the next button it is not always enough to follow the instruction “warmer”. Sometimes the user has to make a small detour to get around a wall or another obstacle. It is expected that these ambiguities make it more interesting to play the game, although they will probably decrease the efficiency and increase the playing time. As game studies have shown, player enjoyment increases if a game is more challenging, and if the players have more control over their actions and more freedom to play the game in the way they want (Sweetser and Wyeth, 2005).

To encourage the users, some exaggerated statements are used when the user is very close to the target (“Feel the heat!”). The instructions given to the user do not prevent the user from triggering an alarm while walking around. As soon as he triggers an alarm he has lost the game.

The Warm/Cold system was based on the same general framework as the Twente system. The same procedures for the generation of referring expressions and for the timing of the instructions were used, only the templates of the sentences were changed.

It was not expected that the Warm/Cold system would perform well in the GIVE Challenge, because in the Warm/Cold system we purposefully generated less than optimal instructions, whereas the GIVE evaluation focused on efficiency and clarity of instructions. The overview of the results of all participating systems in the GIVE Challenge confirms this expectation (Byron et al., 2009).

3 GIVE system results

In this section we discuss the most remarkable evaluation results of our systems, as reported in the overview paper (Byron et al., 2009) discussing the outcomes of the GIVE Challenge. As expected, the Twente server is more efficient than the Warm/Cold system if we look at the objective performance measurements: the task is performed in less time (207.0 vs. 312.2 seconds), using fewer steps (160.9 vs. 307.4). Also the task success rate is higher compared to the Warm/Cold system (35% vs. 18%) (Byron et al., 2009). All of these results are significantly different with $p < 0.05$. In fact, the Warm/Cold system performs significantly worse than all other GIVE systems on these measures.

Most of the questions in the user questionnaire considered the subjective perception of efficiency, in terms of perceived task difficulty, clarity and helpfulness of the instructions. The results shown in the GIVE overview paper (Byron et al., 2009), Figure 10, clearly show that for almost all questions related to efficiency, the results for the Twente system are significantly better than the Warm/Cold system. The only exception is the quality of *referring expressions*, for which the ratings are not significantly different. This is not unexpected since both systems used the same method for the generation of referring expressions. Whenever one of the systems generated a referring expression, the other system would have used the same expression. Unfortunately this also meant that both systems made the same mistakes in the generation of referring expressions. The procedure that calculated the relative position between two objects made an error in some particular cases: in some situations, both systems told the user to press the right button where it should have been the left one and vice versa. Due to this bug it was impossible to win the game without ignoring the given instructions in one of the GIVE game worlds (World 2). In this game world, none of the games with the Twente system was successful, and there was only one player who won the game with the Warm/Cold system, probably because he had prior knowledge of the world (or was extremely lucky).

Byron et al. (2009) show that the players' English skills affected the performance of the systems. Twente is one of the two systems where system performance is not affected by the users' English language skills: it performs equally for players on all skill levels. This may be explained by our adaptive approach: players that did not understand the more complex sentences at the higher levels of our framework, were likely to have slow reaction times and as a consequence the system would have adapted its language use to their skill level. The first level generates very simple sentences that even someone with the lowest level of English could understand. In the Warm/Cold system, all system utterances are very simple, but the system has a slightly complicated introduction text. If a player's English skills were insufficient to understand the introduction he could not be expected to perform well.

4 Warm/Cold: is it more entertaining?

As said the Warm/Cold system takes a more playful approach to the "instruction" giving. Therefore we expected it to have a lower performance than the Twente system in terms of efficiency, but to provide more entertainment. Below we describe how we try to derive information about the NLG systems' entertainment value from the GIVE data, and we compare the results of the Twente and Warm/Cold system in terms of entertainment.

4.1 Measuring entertainment

It is expected that users find a game more interesting if they have to try harder to finally achieve the goal of the game, as is the case in the Warm/Cold system when compared to the Twente system. The GIVE action logs provide some information that may indicate how entertaining the users found each game. First, *cancellation frequency*: if the user is more interested in the game he will be less likely to cancel it. Second, *playing time until cancellation*: if the user does cancel, this is expected to be after a longer period.

As said, the GIVE questionnaire was primarily aimed at measuring clarity and effectiveness of the system's instructions. However, one of the questions can be directly related to the system's entertainment value: if the game is entertaining, the user is more likely to want to play it again. So, in the user questionnaire we expect to find that the score given for *play again* is higher for Warm/Cold than for Twente, even after the user has lost the game.

Finally, we think that if users find a game entertaining, they are at least as interested in the process of playing as in the outcome of the game. Therefore we expect that the more entertaining the users find a system, the less they care about losing. Overall, our prediction is that when the 'gameplay' merely consists of carrying out instructions (as with the Twente system), failing to achieve the task ('losing' the game) will negatively influence the users' subjective judgment of the system, whereas in a more entertaining situation (as with the Warm/Cold system) the users' judgment will be much less influenced by the game's outcome.

4.2 Results

We predicted that when a game is more entertaining, the player is less likely to cancel it. However, the game logs of the Twente and Warm/Cold systems show almost no difference: 25.8% of the games with the Twente system were cancelled, against 24.6% of the games with the Warm/Cold system. We also expected that if a game was entertaining, the users would cancel it after a longer period. However, we found that the mean playing time before a game was cancelled was 234 seconds for the Twente system and 233 seconds for the Warm/Cold system. These results thus contradict our expectation; there is no significant difference between the two systems. The score given for *play again* is not significantly higher for Warm/Cold either (Byron et al., 2009).

We also suggested that when a game is entertaining, the outcome is less important than when it is not. To investigate the extent to which the outcome influenced the subjective ratings of each system, we compared our systems' ratings for the games in which the user won and the games in which the user lost. For each system, we tested the significance of the differences in user ratings between the successful and lost games using Tukey tests. In Table 2 the mean ratings with a significant or near-significant difference are indicated by ** (with $p < 0.05$) or * (with $p < 0.10$). We computed the results presented in Table 2 from the raw evaluation data for our two systems, which were made available to us by the GIVE organizers.

Table 2: Results of the GIVE user questionnaire. Significant differences are indicated by ** (with $p < 0.05$) and * (with $p < 0.10$).

	Twente		Warm/Cold	
Question	Won	Lost	Won	Lost
overall	4.34	4.26	3.93	3.60
task difficulty	**2.15**	**3.83****	3.55	3.57
goal clarity	4.10	3.64*	**3.62**	**2.94****
play again	**2.14**	**3.06****	2.56	2.54
instruction clarity	**4.06**	**3.46****	3.22	2.93
instruction helpfulness	3.64	3.64	3.02	2.91
choice of words	4.22	3.74*	3.89	3.62
referring expressions	**3.96**	**3.33****	3.76	3.36
navigation instructions	3.96	3.76	3.38	3.29
friendliness	3.27	2.94	3.29	3.07
informativity	2.26	2.08	1.67	1.69

For the Twente system, *task difficulty*, *play again*, *instruction clarity* and *referring expressions* show a significant difference between the user ratings, when distinguishing between won and lost games. This shows that losing a game did cause users to judge the Twente system more negatively on these aspects, whereas for the Warm/Cold system no such negative influence of losing was found. This is in line with our hypothesis. However for one question, *goal clarity*, a significant difference between won or lost games was found for the Warm/Cold system, but not for the Twente system. We will try to give an explanation for this in the discussion.

Based on these results, we can neither confirm nor reject our hypothesis that Warm/Cold is more entertaining than the Twente system.

4.3 Discussion

Some of the results presented above differ from what we expected. For example, Table 2 shows a significant difference in *goal clarity* between lost and successful games for the Warm/Cold system, but not for the Twente system. Our hypothesis however was that this should be the other way around. We can explain this because in the GIVE Challenge, the users were led to expect a system aimed at efficiency. The Warm/Cold system has another goal, but this was not, or not clearly, communicated to the user. It seems that the users were confused about the goal of the Warm/Cold game, and "blamed" the explanation after losing a game.

In general, the evaluation results for both systems were probably strongly influenced by the users' expectations. In the introduction of the GIVE game, the NLG system was presented to the user as a 'partner" or 'assistant" who would "tell you what to do to find the trophy. Follow its instructions, and you will solve the puzzle much faster."[2] In short, all information provided to the users suggested that the instructions would be as helpful as possible. The players thus expected a co-operative assistant that would obey the well-known conversational maxims as proposed by Grice (1975). However, we intentionally failed to fulfill some of the maxims to make the Warm/Cold system more challenging. We flouted the Maxim of Manner: our instructions were obscure, and we introduced ambiguity in our direction giving. This also violated the Maxim of Quantity: we gave less information than we could.

Instead of just blindly following the instructions, in the Warm/Cold version the user should think of a strategy to be able to win the game, which we expected would make the game more entertaining. However, the users probably perceived this system behaviour as uncooperative, given that the GIVE users had been told the system would be assisting them, not playing a game with them. This probably explains the lower ratings on all questions for the Warm/Cold system compared to the Twente system.

5 Suggestions for the GIVE Challenge

One important thing that was not taken into account in the GIVE Challenge is the learning effect. In our testing experiments, we saw that players who played the game for the second time performed very well compared to players who played the game for the first time. Even when the game world was different the second time, the experienced players were clearly faster compared to the novice players.

This learning effect could be explicitly taken into account in future editions of the Challenge by having the users always play two games. The results of those games could be compared to gain insights in the learning effects and to compare the NLG systems in terms of their suitability for users with different levels of experience. It would also be possible to only use the results of the second game in the comparison of all systems, to ensure that all users would at least have the same basic level of experience with the game (to prevent the mistakes of "absolute beginners" from dragging down the system results).

To at least partially avoid learning effects during system development, a simple random task generator could be made to vary the order in which the buttons should be pressed to find the trophy. Having a new task available for each round of system tests means that all players, new or experienced, will have to read and follow the instructions to be able to solve the game. They cannot rely on their prior knowledge even if they have played the same world before.

In the GIVE evaluations, three different game worlds were used. Of these, World 1 was quite similar to the development world provided to the Challenge participants to test their NLG systems on. The results of the Challenge clearly show that most NLG systems performed much better on World 1 than on the other two game worlds. To avoid this 'overfitting' effect in the future, it would be good to have not only a random task generator, but also a random world generator that would allow the NLG systems to be tested with more than one world during development.

The virtual environment that was used in this first GIVE Challenge is very simple: the only actions the user can perform are walking, turning and pressing buttons. An obvious next step would be to make the environment more complex and thus more realistic and interesting (for both users and researchers). For example, adding more 'decoration' to the virtual environment would not only make it look nicer but also open possibilities for the generation of more challenging referring expressions. In the current environment a user could only make discrete movements and turns; it would be more natural if the user could make continuous movements. This would also call for more complex, high-level navigation instructions. It would also be more interesting for the users to have more actions available, such as jumping or even shooting in a game-like task, or picking up and manipulating objects in a more serious task, e.g. a construction task where the system functions as a virtual tutor (Kopp et al., 2003).

For future Challenges, a clear choice needs to be made whether the GIVE application should be as a serious one or a game-like one. In the current Challenge, the goal of the system was not communicated very clearly to the users. Mixed signals

[2] http://www.give-challenge.org/about.php

were sent: on the one hand GIVE was described to the users as a game, and the task they needed to carry out was not a very serious one (finding a trophy), while on the other hand the NLG systems were evaluated as if used in a serious application and not a game.

We suggest that it would be more realistic for serious instruction giving systems, aimed at achieving optimal clarity, effectiveness and efficiency, to be evaluated in the context of a serious user task (a so-called 'serious game' application). A variation of the GIVE Challenge could then be created for more entertainment-oriented systems, which should be clearly introduced as such to the user so as to avoid any false expectations. For such an Entertainment Challenge, a game-like task such as this year's trophy-search would be quite suitable. To properly evaluate more playful generation systems, other evaluation measures would have to be used, for example the FUN questionnaire developed by Newman (2005) to evaluate player enjoyment in role playing games. This questionnaire consists of 16 questions, measuring the degree in which (1) the user lost track of time while playing, (2) felt immersed in the game, (3) enjoyed the game, (4) felt engaged with the narrative aspects of the game, and (5) would like to play the game again. For an entertainment-oriented version of the GIVE Challenge, the FUN questionnaire could easily be adapted to a new game context, as was also done by Tychsen et al. (2007).

All in all, this first edition of the GIVE Challenge forms a good basis as well as a source of inspiration for future challenges. We look forward to other successful GIVE Challenges in the coming years.

Acknowledgments

We are grateful to the organizers of the GIVE Challenge for the development of the GIVE framework and for collecting the evaluation data. In particular we thank Alexander Koller for his quick responses to all our questions. This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

References

Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore and Jon Oberlander. 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, special session on Generation Challenges.

H.P. Grice. 1975. Logic and Conversation. *Syntax and Semantics 3: Speech Acts*, Cole et al. (eds.), pp 41–58.

Stefan Kopp, Bernhard Jung, Nadine Leßmann and Ipke Wachsmuth. 2003. Max - A Multimodal Assistant in Virtual Reality Construction. *KI Künstliche Intelligenz*, 4(3): 11–17.

Ken Newman. 2005. Albert in Africa: Online Role-Playing and Lessons from Improvisational Theatre. *ACM Computers in Entertainment*, 3(3).

Penelope Sweetser and Peta Wyeth. 2005. Game-Flow: A Model for Evaluating Player Enjoyment in Games. *ACM Computers in Entertainment*, 3(3).

Anders Tychsen, Ken Newman, Thea Brolund and Michael Hitchens. 2007. Cross-Format Analysis of the Gaming Experience in Multi-Player Role-Playing Games. *Situated Play, Proceedings of DiGRA 2007 Conference*, pp. 49–57.