Playful versus serious instruction giving in a 3D game environment

Roan Boer Rookhuiszen, Mariët Theune*

Human Media Interaction
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

In this paper we introduce two NLG systems that we developed for the GIVE challenge, which was aimed at the evaluation of natural language generation (NLG) systems. The Challenge involved automatically generating instructions for users to carry out a task in a 3D game environment. One of our systems focused on generating optimally helpful 'serious' instructions while the other focused on entertainment, providing more playful instructions. We used the data gathered in the Challenge – both subjective user ratings and objective task performance data – to compare the efficiency and entertainment value of both systems. We found a clear difference in efficiency, but were unable to prove that one system was more entertaining than the other. This could be explained by the fact that the set-up and evaluation methods of the GIVE Challenge were not aimed at measuring entertainment. Based on our experiences, we give some suggestions for the set-up of future installments of the Challenge.

Key words: Instruction-giving, 3D environment, Natural Language Generation, game, evaluation, efficiency vs. entertainment

*Corresponding author. Tel. +31 (0)53 489 4311. Fax +31 (0)53 489 3503.
Email addresses: a.r.boerrookhuiszen@student.utwente.nl (Roan Boer Rookhuiszen), m.theune@ewi.utwente.nl (Mariët Theune)

1. Introduction

In computer games, any language that is directed at the player is usually completely pre-scripted, instead of being generated on the fly during game play. No use is made of the possibilities offered by language generation (NLG) technology to dynamically generate system messages or utterances of non-player characters, and to automatically adapt them to player characteristics or other aspects of the game play situation. In general, NLG has only rarely been used for entertainment-oriented applications, apart from some forays into automatic story generation (e.g., [3]). In this paper we bring together the two fields, which so far have been largely separate: computer games and NLG. Among other things, we show that the evaluation of entertainment-oriented NLG cannot be done in the same way as the evaluation of NLG for serious applications (with the latter being a difficult task in itself, and on-going research topic in NLG).

Natural Language Generation (NLG) is the automatic conversion of some non-linguistic representation of information to written text in natural language, e.g., English [12]. Common applications of NLG are the generation of weather forecasts and various other kinds of reports. Such systems take structured, often numerical data as input (e.g., meteorological data) and automatically generate textual summaries of these data as output.

NLG is also used for the generation of system utterances in various kinds of natural language dialogue systems, which employ spoken or written language to converse with their users. Dialogue system applications range from relatively simple systems for ticket reservation or travel information over the telephone to complex virtual tutors and training systems where the interface takes the shape of one or more embodied agents: virtual humans that interact with the user by means of not only language but also gestures and other nonverbal signals [16]. In dialogue systems, the input for language generation is some formal presentation of what is to be communicated by the system to the user. The job of the NLG component is to convert this input to natural language sentences. Depending on the available communication channels, these sentences can be printed on screen, converted to speech by a subsequent speech synthesis component, and/or enriched with instructions for facial animation and gesture generation (if the system is represented by an embodied agent). The great advantage of using NLG in dialogue systems is that system utterances are not pre-scripted but generated during interaction, and thus can be dynamically adapted to what the user has been saying

or doing. For example, the system might adapt its word choice to that of the user [1, 7]. Such context-sensitivity is difficult to achieve using 'canned' utterances.

As mentioned above, so far almost all work on NLG has been in the context of serious applications. Thus the main goal of NLG systems has been the generation of texts or system utterances that get some information across to the user in an optimal way. Despite its advantages in terms of dynamism and adaptivity, NLG has rarely been used in entertainment-oriented applications such as games. The work we present here is a first step in the direction of employing NLG for more playful applications.

In this paper, we present two NLG systems we developed to take part in a recent shared task evaluation challenge for NLG: Generating Instructions in Virtual Environments (GIVE). This research challenge was developed for the NLG community to provide a new approach to NLG system evaluation [8]. The shared task of GIVE was the generation of instructions for users in a game-like 3D environment (see below for more details). We participated in the GIVE Challenge with one 'serious' NLG system that was focused on generating maximally helpful instructions – the Twente system – and one 'playful' NLG system that was intended to be more game-like and thus entertaining – the Warm/Cold system. The GIVE Challenge was presented as a game to its users, who were invited to "play a game", but the evaluation criteria used in the Challenge focused on effectiveness and efficiency of the generated instructions, not on their entertainment value. In other words, the NLG systems were evaluated as if used in a serious application rather than a game. Nevertheless, in this paper we try to use the data collected in the evaluation period of the GIVE Challenge to compare our two systems in terms of not only efficiency, but also entertainment.

The paper is structured as follows. First, we introduce the GIVE Challenge in more detail in Section 2. Then, in Section 3 we describe the NLG systems we developed. Our hypotheses on the differences between the systems, and the methods to measure those differences are discussed in Section 4. We present the evaluation results of our systems in Section 5, and discuss them in Section 6. We end with a conclusion and pointers to future work in Section 7.

2. The GIVE Challenge

The GIVE Challenge was designed for the evaluation of automatically generated instructions that help users carry out a task in a game-like 3D environment. We participated in the first installment of the Challenge: GIVE-1.[1] This first round was intended as a pilot experiment to investigate the feasibility and challenges of the GIVE evaluation methodology [2]. The experiences with GIVE-1 brought to light some points for improvement, some of which are discussed in this paper. Overall, the first GIVE Challenge was a success and in future years there will more installments.

The GIVE Challenge tries to tackle a difficult problem in the field of natural language generation: evaluation. Since multiple outputs of an NLG system may be equally good – the same information can be expressed in natural language in a variety of ways – it is difficult to automatically evaluate generated texts against a 'gold standard' of texts written by humans. Moreover, automated evaluation metrics do not always correspond to human judgments. Human assessments of generated texts are therefore preferred, but carrying out lab-based evaluation experiments is time-consuming and labor-intensive. In GIVE, an alternative approach is taken: to carry out a task-based evaluation over the Internet. This enables the collection of large amounts of evaluation data, consisting of task performance data and subjective ratings gathered with a user questionnaire [2, 8]. For GIVE-1, a website was set up with some short instructions and the game, which could be played in an ordinary web browser. Players were recruited via (mainly NLG related) email distribution lists and postings on other Internet websites. In total 1143 games were played by people from all over the world, with the largest number of users coming from the USA, Germany and China.

2.1. The task

Users of the GIVE system were asked to perform a task in a game-like 3D virtual environment. To win the game, they had to follow the instructions that the NLG system produced.

The 3D environment presented to the player of the GIVE-game consisted of one or more rooms, connected with doors. There were some objects (e.g. a chair, a lamp) in the world that could be used as landmarks for navigation. On several walls there were square buttons of various colors. The objective

[1] http://www.give-challenge.org/research/page.php?id=give-1-index

Figure 1: Screenshot of the GIVE game, showing the tutorial room where the users learned the controls of the game, before entering the actual game world.

of the GIVE-game for the player was to find a trophy without triggering an alarm. The trophy was hidden in a safe behind a picture on one of the walls. The safe could only be opened by pressing multiple buttons in the right order.

The user had a first person view of the world and could walk through it and turn to the left or the right, but he could not walk through walls and closed doors. The user could also press buttons. The function of each button however was unknown to the user: pushing a button could open a door, move a picture, but also deactivate or trigger an alarm. There was also one floor tile that triggered an alarm if the user stepped on it without having deactivated the alarm first. It was sometimes necessary to press multiple buttons in a specific order to perform one of the actions described above.

Figure 1 shows the interface for the user. At the top of the screen instruction sentences are presented, telling the user which actions he should perform and helping him to achieve the goal. The NLG system generating those instructions had complete knowledge of the world and the actions to be performed in order to win the game; see section 2.2.3.

Three different game worlds were used; for each game one of these was

randomly selected. The worlds had different layouts and provided different levels of difficulty for the instruction-giving NLG system. The first room of each world was a tutorial room where the users practiced with the controls of the game (see Figure 1); this was the same for all worlds and NLG systems.

2.2. Architecture

The goal of the GIVE Challenge was to develop an NLG system for instruction giving; not to implement an entire game architecture. Each participant of the challenge therefore only had to implement the language generation part of the game. All other game components were provided by the GIVE organizers. Below we list the main components of the GIVE game environment; more details on the software architecture can be found in [8].

2.2.1. The client

The client is the actual program the users used to play the game. It could be started from the GIVE website. The client displayed the 3D environment in which a user could walk around and perform several actions. It also displayed the instructions generated by the NLG system. Before and after the game, the client presented the users with a questionnaire; see section 2.3.

2.2.2. The matchmaker

During the evaluation period of the GIVE Challenge (7 November 2008 - 5 February 2009), the GIVE organizers ran a matchmaker server. This server held a list of all NLG systems made by the participants of the challenge. As soon as a user started a client, the matchmaker randomly assigned an NLG system to this client. After the game was finished (with or without success), a complete log of all actions performed by both the NLG system and the user was saved in a database for later evaluation.

2.2.3. The NLG system

The language generation part of the game was implemented by each team participating in the Challenge. The input for language generation consisted of a plan containing the sequence of actions the user should perform to successfully achieve the task (i.e., win the game). This plan was updated after each user action. Furthermore the system had complete knowledge of the virtual environment; it knew the position and properties of all objects in the environment, and which objects were visible to the user from his current position. Based on this information the NLG system generated instruction

sentences telling the user what he had to do. The only feedback on the system's instructions were the actions a user performed after having received the instruction, and a notification whenever the user pressed a 'Help' button.

To develop their NLG system, the participating teams were provided with a development world to test their systems on. The actual worlds used in the GIVE game had the same general properties as this development world, but a different lay-out.

In total 4 teams from the USA, Spain and The Netherlands participated in the Challenge, with 5 different NLG systems [2]. As our contribution we created two NLG systems, which are discussed in this paper.

2.3. Questionnaire

Before and after the game, the user was confronted with an optional questionnaire. This questionnaire was designed by the organizers of the GIVE Challenge; it was the same for each NLG system. Before the game, the user was asked for the following personal information: age, profession, level of computer expertise, level of proficiency in English, and experience playing video games. Then the user played a game, with a randomly assigned combination of game world and NLG system. After the game was finished, the user was asked to rate various aspects of the game experience such as the clarity and helpfulness of the instructions, and the friendliness of the system. The user was also asked to rate the quality of the direction-giving system with an overall score. Most questions had to be answered with a rating on a 5-point scale. The full list of questions asked in the post-questionnaire can be found in Figure 2.

3. 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 a 'classic' NLG system. It 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.

7-point scale items

Overall: What is your overall evaluation of the quality of the direction-giving system? (1 = very bad, 7 = very good)

5-point scale items

Task difficulty: How easy or difficult was the task for you to solve? (1 = very difficult, 5 = very easy)

Goal clarity: How easy was it to understand what you were supposed to do? (1 = very difficult, 5 = very easy)

Play again: Would you want to play this game again? (1 = no way!, 5 = yes please!)

Instruction clarity: How clear were the directions? (1 = totally unclear, 5 = very clear)

Instruction helpfulness: How effective were the directions at helping you complete the task? (1 = not effective, 5 = very effective)

Choice of words: How easy to understand was the system's choice of wording in its directions to you? (1 = very unclear, 5 = very clear)

Referring expressions: How easy was it to pick out which object in the world the system was referring to? (1 = very hard, 5 = very easy)

Navigation instructions: How easy was it to navigate to a particular spot, based on the system's directions? (1 = very hard, 5 = very easy)

Friendliness: How would you rate the friendliness of the system? (1 = very unfriendly, 5 = very friendly)

Nominal items

Informativity: Did you feel the amount of information you were given was: too little / just right / too much

Timing: Did the directions come: too early / just at the right time / too late

Figure 2: The post-game evaluation questions.

3.1. Serious instructions: the Twente system

The organization of the GIVE Challenge provided all participating teams with an example implementation of an NLG system. This system was very basic and told the user to perform only one action at a time, leading to sequences of instructions such as "Take one step forward. Take one step forward. Take one step forward. Turn left." We carried out some user tests with this example system, which revealed that the simple instructions were easy to understand, especially for first-time users; however they were very annoying for more experienced users, who had already played the game once or twice.

In our first attempt at implementing our own NLG system we therefore combined all steps to be taken before making a turn, plus the turn itself, into one instruction. For example, "Walk forward 3 steps and then turn left." More experienced users did perform better with this new system than with the example system: they used less time to reach a button, and found the instructions clearer. However, we found that first-time users could not handle the increased complexity of the instructions. 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 global instructions that are expressed using more complex sentences. Some example sentences generated by the different levels are the following.

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 blue button."

Level 3: Also a combination, but only referring to objects when the user can see them: "Walk towards the blue button and press it."

At the third 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. See Figure 3 for an example.

The instructions on all levels are generated using the same general framework. Certain actions to be performed by the NLG system are the same for all levels: interpreting user actions and other events, the generation of referring expressions ("The blue button"), the check whether users are performing

Figure 3: Example of a level 3 instruction in the Twente system. The 'buttons' are the differently coloured cubes on the walls.

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 after the player has fulfilled the previous, complex instruction. At the third level, the system first gives an instruction indicating 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; see the Appendix for an example.

On all levels, sentences are generated in a similar way, using small templates. The templates are different between levels, but they all just need the referring expressions, directions and the number of steps to be filled in. For example, "Walk forward [N] steps and then press [button]". 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 blue button"). If another visible button happens to have the same color, the relative position

is mentioned ("The second blue button from the right").

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

We started the implementation process by implementing the first level. Based on this first implementation the other levels were created. When we had implemented a first version of each level, we asked a few users to play the game at each of the levels. During these tests, we received suggestions for several small adaptations. For example, in our first version of level 2 a sentence consisted of a turn instruction followed by a walk instruction. For example,"Turn right then walk 3 steps". People found it more natural and easier to understand when the navigation plan was segmented differently, so that combined instructions always started with a walk instruction followed by a turn, e.g., "Walk 2 steps then turn right."

The final version of the Twente system, as used in the GIVE Challenge, was made to be *adaptive*: the NLG system tries to automatically fit the level to the user's needs. Unlike in our test games, in the actual GIVE Challenge we had no way of knowing whether the users were new or experienced. Some users might have played the game before (probably using a different NLG system) but this information was not available to our system. Also, we expected that novice users would learn while they were playing the game, so their level of experience would not be fixed throughout the game. The Twente system therefore tries to automatically detect the level of experience of the user by his game play, and changes the level of instructions accordingly. This is done as follows. When the game starts the level used is 2. Every second, the system checks the number of actions the user performed in the last 5 seconds. When this number exceeds a certain threshold, this is taken as a sign of experience. The system assumes that 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, the user presses the 'Help' button or moves in the wrong direction.

Different values are used as thresholds for lowering or raising the instruc-

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

tion level between levels 1 and 2, and between levels 2 and 3. 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 higher than the given threshold, and decreased when the number of actions in the last 5 seconds is lower than the given threshold. We have determined the thresholds by letting people play the game at a fixed level for several times. We clearly saw a learning effect: players were much faster the second time they played the game.

In general, we have set fairly high thresholds in order to prevent too frequent switching between levels. For example, the higher levels are only reached if the user is playing relatively fast, performing at least 1 action per second. However, in the higher levels the user also has to read more text because the instructions are longer, resulting in the user performing no actions for some time while reading. The level should not be switched down again immediately when this happens and therefore the level is only lowered if the user's action rate drops very low. In the Appendix a part of a game run with the Twente system is given, showing a transition from level 2 to level 3 and back again.

3.2. Playful instructions: the Warm/Cold system

To make the task more interesting for the users, in our second NLG system we created a more game-like situation, where the system tries to simulate a warm/cold game. Instead of telling the user exactly what to do, the only instructions given are variations on "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. To encourage the users, some exaggerated statements are used when the

Figure 4: The Warm/Cold system tells the user he is getting "Warmer" (i.e., closer to the next target button).

user is very close to the target (e.g., "Feel the heat!"). In the Appendix we show a part of a game run with the Warm/Cold system, with more examples of system utterances.

Most of the utterances generated by the Warm/Cold 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 4: 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. Also, 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. It is expected that these ambiguities and risks 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 [15].

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, and there are no levels.

We did not expect the Warm/Cold system to perform well in the GIVE Challenge, because it purposefully generates 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 confirmed this expectation: the Warm/Cold system, being the only 'non-serious' submission, was among the worst-performing systems on all evaluation measures [2].[2]

4. Evaluation

To test if our NLG systems achieved their respective goals of being efficient or entertaining, we evaluate them using the data collected for our systems in the GIVE Challenge. These include the action logs of the system and the answers to the questionnaires. We compare the results of the Twente system and the Warm/Cold system in light of their two goals. Our main hypotheses are:

1. The Twente system is more efficient than the Warm/Cold system.
2. The Warm/Cold system is more entertaining than the Twente system.

[2]The three other NLG systems taking part in the Challenge were all 'serious' systems that focused on providing optimally helpful instructions. The first system, developed at Union College NY, was similar to the Twente system in that it automatically adapted its instruction strategy to the user. By default, it used a landmark-based strategy ("Go to the chair, and then turn right") but it switched to a simpler path-based strategy ("Go 5 steps forward") if the user pressed the 'Help' button or did not make any progress toward the target object [14]. The second system, developed at the Universidad Complutense de Madrid, focused on the generation of global instructions such as "Go to the next room", referring to high-level spatial concepts such as rooms and corners [5]. The third system, developed at the University of Texas, was similar to the example system provided by the GIVE organizers, except that it grouped steps together, thus generating instructions similar to level 1 of the Twente system [4].

To test these hypotheses, the only available information is the data collected during the evaluation period of the GIVE Challenge. A disadvantage is that the evaluation questions used in the Challenge were about clarity and task performance rather than the users' experiences. This means the results of the questionnaire are suitable for testing Hypothesis 1, but less so for testing Hypothesis 2, for which we mostly have to rely on indirect clues instead. Below, we describe how we measure the efficiency and entertainment value of our NLG systems in terms of the data available from GIVE.

4.1. Measuring efficiency

The efficiency of a system can be measured objectively by using the logged performance results. One system is more efficient than another if using this system, the users successfully performed the task in *less time* and with *less detours* (i.e., using fewer steps) than when using the other system. We also take *task success rate* as an objective indicator of efficiency.

Most questions in the post-questionnaire (Figure 2) deal with the subjective perception of efficiency. We assume that one system is perceived as more efficient than another if it scores better on *task difficulty*, *goal clarity*, *instruction clarity*, *instruction helpfulness*, *choice of words*, *referring expressions*, *navigation instructions*, *informativity* and *timing*.

Also the overall rating of the quality of the direction-giving system *(overall)* is expected to be better, based on the assumption that the users mostly based this rating on the clarity and helpfulness of the instructions, rather than on the entertainment value of the game.

4.2. 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 and have more freedom to choose their own actions, 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 is 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. Cf. Pagulayan et al. [10], who state that in games, unlike what they call 'productivity applications', the process is more important than the outcome. We therefore assume that the more entertaining the users find a system, the less they care about losing. Overall, our prediction is that when the 'game-play' 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.

5. Results

The results presented in this section are based on the data gathered in the GIVE Challenge. The subjective user ratings for the Twente and Warm/Cold systems, and some of the objective measures, were taken from the official report discussing the outcomes of the Challenge [2]. We computed the other results from the raw evaluation data for our two systems, which were made available to us by the GIVE organizers. Before we present the evaluation data and discuss the results in the light of our hypotheses, we first provide some information on the users who played the game with the Twente or the Warm/Cold system.

5.1. Participants

In total, 214 games were played with the Twente system and 269 with the Warm/Cold system. However, since filling in the pre- and post-game questionnaires was optional, questionnaire results are only available for a portion of all games that were played.

Roughly one third of the games were played from IP addresses in the USA, another third from Germany and the rest from other countries. Around 80% of the games were played by male users, 10% by female users and for 10% of the games, the users did not specify their gender. Unfortunately we were unable to determine whether all games also represent different users, as the GIVE Challenge only distinguished between game plays, not between users.

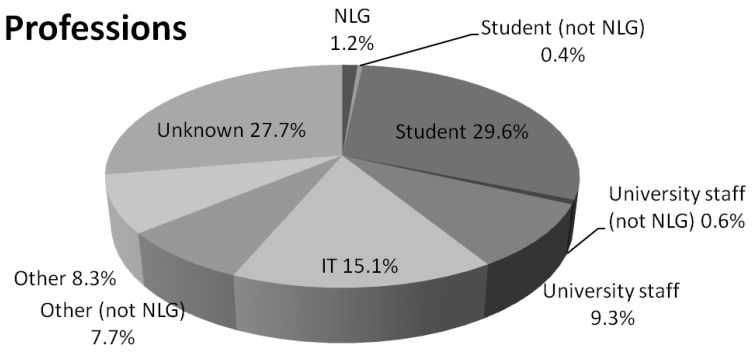

Figure 5: Distribution of the professions of the users.

It is possible that users played a game with another NLG system before they used one of our systems.

Users were recruited via various websites and mailing lists. As mentioned in Section 2, some of those were NLG related. It would be interesting to know how many of our users had a background in NLG, since they might have different expectations from an NLG system, compared to users without NLG knowledge. Unfortunately, the pre-game user questionnaire contained no questions concerning the user's level of NLG experience. The questionnaire did contain an open question asking for the user's profession, but only relatively few answers unambiguously showed whether the user had an NLG related job or not. As we can see in Figure 5 we only know for sure that 1.2% of the users have an NLG related job, while 23.8% are definitely not involved in NLG. Of the latter group, 15.1% work in IT, and 8.7% are students, university staff or people in other jobs that are not NLG related. However, the level of NLG experience of the remaining 75% of the users is unknown. Of these, 27.7% did not provide any information about their profession, while the others only provided vague descriptions such as 'student' or 'researcher'.

5.2. Hypothesis 1: efficiency

In Table 2, the results from the GIVE questionnaire are reported as the mean ratings given by the users of each system. Each mean value was calculated from roughly 50 answers. Significance was tested by using Tukey tests; the means that are significantly different (with $p < 0.05$) are shown in bold face. As we have seen in Section 4.1, most of the questions in the questionnaire consider the subjective perception of efficiency. The results in

Table 2: Results of the GIVE user questionnaire, taken from [2]. Results that are **significantly different** (with $p < 0.05$) are given in bold face. For *informativity* and *timing* we give the percentages of "just right" answers; these were computed for successful games only.

Question	Twente	Warm/Cold
overall	**4.3**	**3.6**
task difficulty	**4.0**	**3.5**
goal clarity	**3.9**	**3.3**
play again	2.4	2.5
instruction clarity	**3.8**	**3.0**
instruction helpfulness	**3.6**	**2.9**
choice of words	**4.1**	**3.5**
referring expressions	3.7	3.5
navigation instructions	**4.0**	**3.2**
friendliness	3.1	3.1
informativity	51%	51%
timing	60%	49%

Table 2 clearly show that for all questions related to efficiency except *referring expressions* there is a significant difference between the means of the two systems, with the Twente system consistently scoring higher.

The fact that the *referring expressions* rating was not significantly different between the two systems is not unexpected, since they both used the same method for referring expression generation. As a consequence, both systems used the same expressions to refer to the same buttons. Unfortunately this also meant that both systems made the same mistakes. The procedure that calculated the relative position between two objects made an error in some particular cases: in some situations, the user was told 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).

The objective measurements also confirm that the Twente system is more

efficient than the Warm/Cold system: the task was performed in less time (207.0 vs. 312.2 seconds) and using fewer steps (160.9 vs. 307.4). Moreover, the task success rate was significantly higher (35% vs. 18%) [2].

An interesting result reported in [2] is that the players' level of English affected the performance of most NLG systems in terms of task success, meaning that for these systems, the percentage of successful games was significantly lower for players with only basic English skills. There were only two NLG systems that were not affected by this factor and showed an equal performance for players on all skill levels. One of those was the Twente system. 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. As a consequence the system would have lowered the instruction level, thus basically adapting its language use to the players' English skills. 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 had a slightly complicated introductory text. If a player's English skills were insufficient to understand the introduction, he could not be expected to perform well, in particular if he was not familiar with the concept of a Warm/Cold game (which may have been the case for players with a non-Western cultural background).

For other player characteristics, no significant influences were found.

All in all, and not very surprisingly, Hypothesis 1 is confirmed by the evaluation data: the Twente system is more efficient than the Warm/Cold system.

5.3. Hypothesis 2: entertainment

In relation to our second, more interesting hypothesis, we predicted that when a game is more entertaining, the player is less likely to cancel it. However, the game logs 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 entertaining games would be cancelled after a longer period. However, the mean playing time before cancellation was 234 seconds for the Twente system and 233 seconds for the Warm/Cold system. These results contradict our expectation; there is no significant difference between the two systems. The scores for *play again* are not significantly different either (see Table 2).

Table 3: Results of the GIVE user questionnaire for won versus lost games. 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

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 between the means of the successful and lost games by using Tukey tests. In Table 3 the means with a significant or near-significant difference are indicated by ** (with $p < 0.05$) or * (with $p < 0.10$).

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 Hypothesis 2.

6. Discussion

Some of the results presented in the previous section differ from what we expected. For example, Table 3 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 'serious' system aimed at efficiency. In fact, all NLG systems in the GIVE Challenge were aimed at efficiency, except the Warm/Cold system. The Warm/Cold system had another goal, entertainment, 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 of the goal after losing a game.

In general, the evaluation results for both the Twente and the Warm/Cold system 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." In short, all information provided to the users suggested that the instructions would be as helpful as possible. The players thus expected a cooperative assistant that would obey the Cooperative Principle proposed by the philosopher Grice: "Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose of the talk exchange in which you are engaged." ([6], p. 45). In accordance with the Cooperative Principle, Grice proposed four conversational maxims [6]:

- Maxim of Quantity: Make your contribution as informative as needed, but not more informative than required.
- Maxim of Quality: Do not say what you believe to be false or for which you lack adequate evidence.
- Maxim of Relation: Be relevant.
- Maxim of Manner: Be perspicuous, and avoid ambiguity.

These maxims can be seen as rules a cooperative speaker uses in a conversation. They underlie most work in NLG, and we have obeyed them for the instructions generated by the Twente system. In contrast, we intentionally

failed to fulfill some of the maxims to make the Warm/Cold system more challenging. We flouted Grice's Maxim of Manner: our instructions were not perspicuous but obscure, and we introduced ambiguity in our direction giving. This is also in violation of the Maxim of Quantity: we gave less information than we could. This made it much harder for the users to understand the direction giving instructions of the system. 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.

Note that the conversational behavior of the Warm/Cold system could still be seen as cooperative, in the sense that its instructions were "such as is required (...) by the accepted purpose of the talk exchange" ([6], p. 45) *if* this purpose were defined as achieving entertainment. However, as mentioned above, the users of the GIVE game were told that the system would be assisting them, not playing a game with them. In other words, the accepted purpose of the talk exchange was to guide the users as efficiently as possible to the trophy, not to entertain them. Given this purpose, the users probably perceived the behavior of the Warm/Cold system as uncooperative, which may explain the lower ratings on all questions for the Warm/Cold system compared to the Twente system.

Finally, it is possible that users with a background in NLG were inherently biased in favor of the more serious NLG systems, including the Twente system, and had a more negative attitude toward the less conventional Warm/Cold system. However, we could not check this, since we were only able to establish for 1.2% of the users that they definitely had previous experience with NLG systems; see Section 5.1. In future GIVE Challenges, it would be useful to include a question on the user's NLG experience in the user questionnaire.

7. Conclusions and Future Work

The GIVE Challenge was set up as a shared task for the evaluation of NLG systems, measuring mostly efficiency-related quality aspects of automatically generated instructions. We participated in the GIVE Challenge with two NLG systems. In this paper, we used the evaluation data gathered in the GIVE Challenge to compare the efficiency and the level of entertainment provided by our systems' instructions. The results clearly showed that our 'serious' NLG system, the Twente system, was the most efficient. How-

ever, we found no conclusive evidence for the hypothesis that our Warm/Cold system, which was designed to be playful, would be perceived as more entertaining than the Twente system. The latter finding may be at least partially explained by the fact that the GIVE data were not fully suitable to measure entertainment, since this was not the focus of the GIVE organizers who designed the questionnaire. In addition, the purpose of the GIVE game was not clearly communicated to the users. Mixed signals 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. For future Challenges, we think a clear choice needs to be made whether the GIVE application should be a serious one or a game-like one.

In addition, in future installments of the GIVE Challenge, it would be good if the participating teams could adapt the user questionnaire to their own research questions. In our case, this would allow us to use better methods for measuring entertainment value, such as the FUN questionnaire developed by Newman [9]. This questionnaire was designed to evaluate player enjoyment in roleplaying games, 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. The FUN questionnaire looks like a good starting point for our evaluation purposes, and could easily be adapted to our own game context, as was also done by Tychsen et al. [17]. Another potentially useful questionnaire is the Game Experience Questionnaire, which measures game experience along dimensions such as flow, immersion, affect and competence, as identified by [11].

The results of the GIVE Challenge were presented at the ENLG 2009[3] workshop in Athens, and a meeting was held there to discuss the future of the GIVE challenge [13]. This meeting was attended by the organizers and participants of GIVE-1 and other interested parties. The main suggestion made to improve GIVE-2, the next installment of the GIVE Challenge,[4] was to move from discrete to continuous worlds, i.e., worlds in which the player can move around freely rather than in discrete steps. This will result in a more interesting task from the point of view of NLG, since it will no

[3]The 12th European Workshop on Natural Language Generation.

[4]http://www.give-challenge.org/research/page.php?id=give-2-index

longer be possible for the NLG systems to generate very simple instructions in terms of steps to be taken, as is done at levels 1 and 2 of our Twente system (Section 3.1). Instead, it will be necessary to generate more realistic, high-level instructions referring to landmarks and spatial concepts such as rooms, doors and corridors, cf. [5].

Other suggestions for improvement were related to the nature of the GIVE task and the way it is advertised to the users. Apparently, user feedback showed that some users who expected a game were disappointed when they found out that GIVE was not really a game. Various ideas were put forward to add more game elements to the GIVE set-up. For example, multiple player tasks could be introduced and players might be rewarded with points for achieving them. More frequent rewards might lead to more fun in playing and encourage users to play multiple games, as they will probably try to get as many points as possible. Similar ideas were the introduction of a timer, encouraging the users to play against the time, and the addition of a secondary task (for example, collecting items in the game) with which the player could win a real-life reward. Other suggested improvements to the GIVE client, such as more landmarks, visually more diverse worlds or themed worlds, might further help to increase the commitment of the players to the game.

The suggestions for making GIVE more game-like are not yet implemented in GIVE-2, but it is interesting to see that there is a tendency to move in this direction. The GIVE Challenge is currently still focused on the evaluation of 'serious' NLG systems, but by having experimented with a more entertainment-oriented approach and having identified the limitations of the evaluation methods used in GIVE-1 we hope to have paved the way for making future GIVE installments more suitable for the evaluation of playful systems. This will facilitate further research into the use of NLG in games and game-like applications, investigating the different ways how NLG can contribute to entertainment and how this can be evaluated.

8. Acknowledgements

We thank Michel Obbink for his contribution to the development and testing of the Twente system. We are also grateful to the organizers of the GIVE Challenge for developing the framework and 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

[1] Buschmeier, H., Bergmann, K., Kopp, S., 2009. An alignment-capable microplanner for Natural Language Generation. In: Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009). Athens, Greece, pp. 82–89.

[2] Byron, D., Koller, A., Striegnitz, K., Cassell, J., Dale, R., Moore, J., Oberlander, J., 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). In: Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009). Athens, Greece, pp. 165–173.

[3] Callaway, C., Lester, J., 2002. Narrative prose generation. Artificial Intelligence 139 (2), 213–252.

[4] Chen, D., Karpov, I., 2009. The GIVE-1 Austin system. In: Online Proceedings of the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE-1). Http://www.give-challenge.org/research/files/GIVE-09-Union.pdf.

[5] Dionne, D., de la Puente, S., León, C., Gervás, P., Hervás, R., 2009. A model for human readable instruction generation using level-based discourse planning and dynamic inference of attributes. In: Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009). Athens, Greece, pp. 66–73.

[6] Grice, H., 1975. Logic and conversation. In: Cole, P., Morgan, J. (Eds.), Syntax and Semantics 3: Speech Acts. Academic Press, New York, pp. 41–58.

[7] Janarthanam, S., Lemon, O., 2009. Learning lexical alignment policies for generating referring expressions for spoken dialogue systems. In: Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009). Athens, Greece, pp. 74–81.

[8] Koller, A., Byron, D., Cassell, J., Dale, R., Moore, J., Oberlander, J., Striegnitz, K., 2009. The software architecture for the first Challenge on Generating Instructions in Virtual Environments. In: Proceedings of the Demonstrations Session at EACL 2009. Athens, Greece, pp. 33–36.

[9] Newman, K., 2005. Albert in Africa: Online role-playing and lessons from improvisational theatre. ACM Computers in Entertainment 3 (3).

[10] Pagulayan, R. J., Keeker, K., Wixon, D., Romero, R. L., Fuller, T., 2003. User-centered design in games. In: Jacko, J., Sears, A. (Eds.), The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. Lawrence Erlbaum Associates, Mahwah, NJ, pp. 883–906.

[11] Poels, K., de Kort, Y., IJsselsteijn, W., 2007. "It is always a lot of fun!": exploring dimensions of digital game experience using focus group methodology. In: Proceedings of the 2007 Conference on Future Play. Toronto, Canada, pp. 83–89.

[12] Reiter, E., Dale, R., 2000. Building Natural Language Generation Systems. Cambridge University Press, Cambridge.

[13] Striegnitz, K., Koller, A., 2009. Summary of the GIVE meeting at ENLG 2009, unpublished report.

[14] Striegnitz, K., Majda, F., 2009. Landmarks in navigation instructions for a virtual environment. In: Online Proceedings of the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE-1). Http://www.give-challenge.org/research/files/GIVE-09-Union.pdf.

[15] Sweetser, P., Wyeth, P., 2005. Gameflow: A model for evaluating player enjoyment in games. ACM Computers in Entertainment 3 (3).

[16] Traum, D., Fleischman, M., Hovy, E., 2003. NL generation for virtual humans in a complex social environment. In: Papers from the AAAI spring symposium on Natural Language Generation in Spoken and Written Dialogue. pp. 151–158.

[17] Tychsen, A., Newman, K., Brolund, T., Hitchens, M., 2007. Cross-format analysis of the gaming experience in multi-player role-playing games. In: Situated Play, Proceedings of DiGRA 2007 Conference. pp. 49–57.

Appendix

Here we show two (partial) game runs, taken from games that were played for illustration purposes. The first illustrates the differences between two levels of the Twente system and the second shows some instructions given by the Warm/Cold system. User actions are given in small caps, while system messages are given between quotation marks.

Twente system

Time	Actions / System messages	Description
0:22	"Turn left."	*The user started at level 2*
0:22	TURN_LEFT	
0:23	"Walk forward 4 steps then turn right."	
0:23	MOVE_FORWARD	
0:23	MOVE_FORWARD	
0:24	MOVE_FORWARD	
0:25	MOVE_FORWARD	
0:25	"Turn right."	
0:25	TURN_RIGHT	*Level up to 3, because in the*
0:25	"Walk towards the right blue button and press it."	*last 5 seconds more than 8 actions were performed*
0:26	MOVE_FORWARD	*(only the last 6 are shown)*
0:27	TURN_RIGHT	
0:28	MOVE_FORWARD	*The user should find the*
0:30	MOVE_FORWARD	*intended button on his own.*
0:30	TURN_LEFT	
0:32	MOVE_FORWARD	
0:33	TURN_LEFT	
0:34	MOVE_FORWARD	
0:34	"Oops, you're moving in the wrong direction."	*The user moved in the wrong direction and is*
0:35	"Turn right."	*switched back to level 2.*

Warm/Cold system

Time	Actions / System messages	Description
0:24	MOVE_FORWARD	
0:25	"A little warmer"	
0:26	MOVE_FORWARD	
0:27	"A little warmer"	*After each move action*
0:28	MOVE_FORWARD	*a new 'hint' is given.*
0:29	"Warmer"	
0:29	TURN_RIGHT	
0:30	MOVE_FORWARD	
0:30	"Warmer"	
0:31	TURN_LEFT	
0:32	MOVE_FORWARD	
0:33	"Just a little colder"	*The user should make a*
0:33	MOVE_FORWARD	*small detour to find the*
0:34	"Just a little colder"	*intended button.*
0:35	TURN_RIGHT	
0:36	MOVE_FORWARD	
0:37	"Warmer"	
...		
0:49	MOVE_FORWARD	
0:50	"Almost on fire..."	
0:51	MOVE_FORWARD	*Exaggerated sentences are*
0:52	"Feel that heat?! Turn!"	*used when the user is very*
0:53	TURN_LEFT	*close to the button.*
0:54	"Press the blue button."	