Generating and Evaluating Coarse-Grained Instructions in a Virtual Environment

Miguel Martínez Soler and Christian Cossio Mercado

Laboratorio de Investigaciones Sensoriales (LIS) - CONICET
Instituto de Neurociencias, Hospital de Clínicas, UBA
Buenos Aires, Argentina
miguelmsoler@gmail.com,cgcm@rucatech.com.ar

Abstract. We present this work aimed at generating instructions in a 3D virtual environment, as part of a *treasure hunt* game. To achieve this goal we generated natural language text with the objetive of guiding a human user on *where to go* and *what to do* within the virtual world in order to get to a hidden treasure. Our approach, which gives coarse-grained instructions (e.g. `Go to the red room`) showed better results compared with a step-by-step guiding (e.g. `Turn left`, `Go straight`, `Turn right`, `Go straight`).

Keywords: Natural Language Generation; Virtual environment; GIVE Challenge

1 Introduction

There is a need to develop systems capable of generating natural language, in particular with the resurgence of Dialogue Systems [7] mainly due to the newly available technologies in Speech Recognition [9] and Synthesis [8]. This is the area of work of the Natural Language Generation (NLG) field [6].
To evaluate NLG Systems it is necessary to count with environments to test and benchmark different techniques and algorithms. This is where the GIVE Challenge [4] gets into the game, in order to help to evaluate and develop new NLG systems.
In this work we defined a system for the generation of instructions for a *treasure hunt* game environment and implemented within the GIVE platform.
In Section 2 we start giving a brief introduction about the GIVE Challenge and its virtual environment. Section 3 includes details about our implementation and summarizes the main changes with respect to the base *instruction giver*. A selection of experiencies registered during testing of our system are detailed in Section 4. In Section 5 there is more information on the results of our system versus the base GIVE system. Conclusions and possible future work are detailed in Section 6.

2 The GIVE Challenge

In the GIVE Challenge [3, 2, 5] a human user participates in a *treasure hunt* game within a 3D virtual environment, known as a GIVE World. The objective of the system is to generate real-time, natural language instructions that will guide the users to the successful completion of their task [5]. In a GIVE world there are one or more rooms containing objects the user can interact with (e.g. buttons) as well as other items (e.g. lamps, plants and chairs).
The objective of the game is to get to a trophy which can be located in any of the rooms. To do that, for example, the user would need to move through the rooms of the world and push buttons in order to open doors and deactivate alarms, to finally take the trophy. We consider that each room could be identified according to one or more of the following attributes: color (if any), contained objects, absolute position in the world, relative position with respect to the user and the knowledge of whether it was previously visited.

When starting the game there is a tutorial room where the users learn how to interact with the system. After that, the user starts playing with a given GIVE world. The game ends successfully when the user gets the trophy, placed in a safe behind a picture, while it ends unsuccessfully when the user triggers an alarm, by stepping into an area where an alarm was still active, or when the user cancels the game.

The GIVE software performs a discretization of the world, splitting each room in several regions that may contain objects. It also has a planner that computes a detailed plan, given as a sequence of actions that the user should perform in order to get the throphy. This actions have the same discretization level as the GIVE world, that is, each action represents either an action to be performed on an object or a movement to another region. The baseline system considered in this work generates one instruction per action in the plan.

3 Generating Coarse-Grained Instructions

Given the discretization and plan inference implemented in the baseline system, we could implement coarse-grained instructions by *leaving tacit* [1] some of the actions in the plan. For example, when there is a need to go to another room, while the baseline system generates step-by-step instructions (i.e., `Go straight`, `Turn left`, `Push the red button`, `Turn right`, `Go straight`, `Go through the door`, `Go straight`) our system gives less and more significant instructions (i.e., `Push the red button to the left`, `Go to the blue room`).

In our system it is necessary to verbalize actions only if they involve either a room change or an action that applied to an object (i.e., `Push the button` or `Take the throphy`). With this goal in mind, we modified the baseline system to generate instructions for each move action that would involve a room change as well as an instruction for each non-move action.

These modifications were implemented as a algorithm that reads the current plan and returns the first action on it that represents one of the situations

described previously. The algoritm is executed every time the system replans because the user did not follow the given instructions. Thus, when giving an instruction to the user, the system does not verbalize the next action of the plan but the one that represents a room change or a specific action on an object in the world. The algorithm is called also every time the execution of a right action is verified, since it is necessary to calculate the next expected action and generate the corresponding instruction.
After testing our algorithm on a GIVE world we realized that a change in the generation of referring expressions for rooms and objects was necessary as a result of the implemented change in granularity. For buttons, for example, the referring expressions generator supposed that the user was in the region containing it, and therefore, verbalized `Press the thing` quite frequently. This issue was solved by forcing the system to generate the referring expression in any case, even though the user was in the region of the object that had to be manipulated. Additionally, in order to describe rooms we decided to reference them by their color and some object that was present on it.

The solution adopted for referring expressions worked fairly well for the purposes of this work, but in some situations there were some vague descriptions. Particularly, the descriptions of the rooms worked poorly as arised when evaluating the system in the a GIVE test world, where at one point we had to move to a room that was described as "...the room with the flower", as seen in Figure 1, but several rooms matched that description [1]. Then we decided to

Fig. 1. A flower used a landmark of two contiguous rooms

improve our strategy by selecting the object associated with the room by its

[1] In one of the GIVE test worlds by default there is a reference to a flower that is contained in two contiguous rooms.

rarity (i.e. choosing the less frequent object in the world that was present in the room).

4 Testing and enhancing the system

When evaluating the system with the GIVE Challenge 2010 testing worlds there were some situations that led to further analysis and enhancements in our system.

In a certain situation there was a reference to an alarm (see Figure 2) as a way to denote the next room to go. This is useful when the alarm is activated, but when it is not, it is not so easy to distinguish the alarm from the floor, because an alarm is represented by a floor tile that changes its colour when that alarm is activated. Thus, we decided to avoid referring expressions that included an alarm in their content.

We achieved this by defining a *black list* of objects that could not be used as land-

Fig. 2. A alarm used as landmark to reference to a room

marks. We also included doors, buttons and the trophy to the list, in order to prevent the algorithm of generating things like `Go to the room with the door`, `Go to the room with the button` or `Go to the room with the throphy`. In general, the throphy is not visible until the very end of the game, when the user just has to take it. The buttons could be excluded of the list if their color were taken into account (e.g., `Go to the room with the blue button`), but we think that it would still produce ambiguous descriptions.

In some situations there were several buttons with the same color in the same room, and the user have to press only one of them. This problem arises when the button are on the same side of the room, since it is not possible to infer to which one the system is referring to, as seen in Figure 3. This kind of situations

did not happen in the baseline system, because in order to generate a referring expression for that button, the user had to be in the region that contained it. The jump to a higher granularity level brought with it some ambiguity in the description of objects.

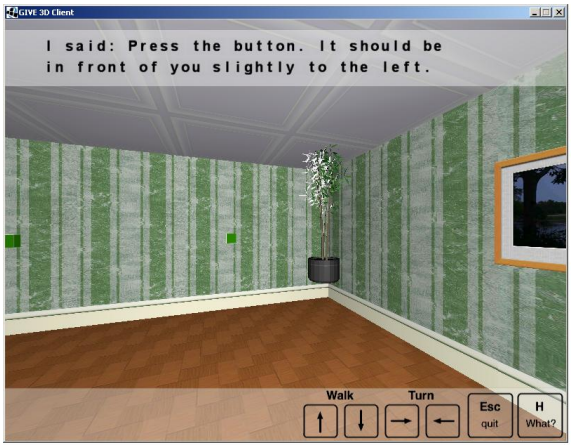

Fig. 3. Ambiguous referring expression for a button

In one of the GIVE worlds that we tested our system, there were some contiguous rooms with no color nor objects. In these cases the system only asks to go to the next room, since there are no properties to refer to. It is ambiguous when there are more than one room to go, as could be seen in Figure 4.

There was a particular problem with a room that was not rectangular. The system asked to press a button that was in front of the user, but the button was in fact on the other side of the wall (see Figures 5 and 6, where the box in the latter denotes user's position and the circle marks the button to be pressed). In these situations the attribute that we considered, i.e. the position of the objects and user visual orientation, were not enough.

5 Comparing results

In this section we analyze how the strategy of our system impacts on the subjective evaluation of the system, according to subjective metrics defined for the GIVE Challenge [5], comparing it with the strategy used by the baseline system. In order to obtain such metrics, we asked a person not related to the NLG subject and to this project to play the game on three diferent worlds and to score each question with a value from -100 (totally false) to 100 (totally true).

As depicted in Table 1, being best values in boldface, according to the average of the surveys of our approach versus the baseline, our system outperforms the baseline system. In that sense, the diferences in questions Q2,

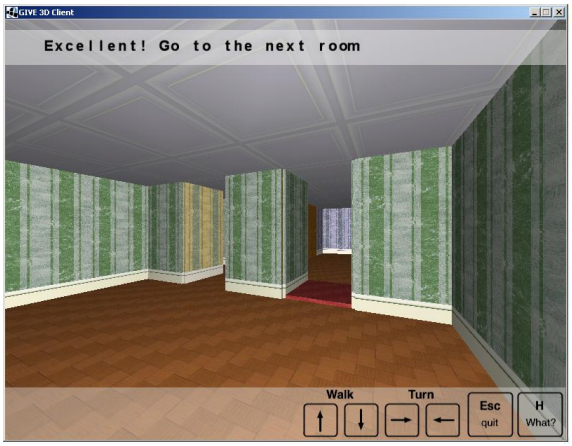

Fig. 4. A reference to an ambiguous 'next room'

Fig. 5. The button to be pressed is on the other side of the wall

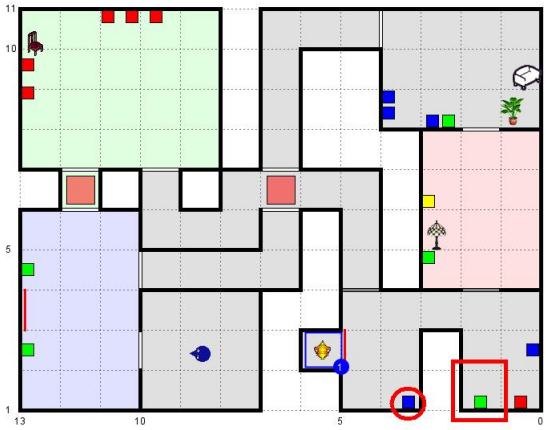

Fig. 6. Map of a world in GIVE

Q4, Q8, Q12, Q13, Q14, Q19 and Q21 are quite significant. It must be noticed, however, that according to metric Q11 the strategy used by the baseline system outperforms ours. This is due to the instructions of the baseline system are very specific (fine-grained), so it gives feedback quite often (like `Wait, that's not what I wanted you to do. I need to make a new plan`). On the contrary, our system rarely makes new plans, because the instructions given are coarse-grained and the user has to make a mistake in the same (high) level in order to force the system to replan (e.g. more to a room different to the one that was specified).

6 Conclusions

In this work we developed a NLG system within the GIVE platform. Our aim was to give instructions of coarse granularity, i.e. giving instructions for *move* actions only for room changes, as well as *push* and *take* actions.
After modifying the baseline system we found several issues trying to adapt the referring expressions, since they were made for fine-grained instructions and contained syntax and words related to that level of granularity.
We were able to get a higher quality solution by generating referring expressions using low-frequency landmarks. We think that this method could be enhanced through several ways, e.g. taking into account those situations where a room is full of objects of a certain type (e.g. lamps) that aren't present in any other room, that currently would not be marked as a 'good' attribute. That is because, though a common item within the world, the object stills being a good landmark since it references the room unequivocally.
With respect to the referring expressions for buttons, we kept the description based on the relative position of the user. However, this approach is not so

Table 1. Average subjective evaluation for Baseline and Ours systems across worlds

#	Question	Base	Ours
1	The system was very friendly	7	**27**
2	The system gave me a lot of unnecessary information	60	**-60**
3	I had to re-read instructions to understand what I needed to do	**47**	60
4	Interacting with the system was really annoying	60	**7**
5	The system's instructions were clearly worded	**7**	0
6	I was confused about what direction to go in	33	**7**
7	I enjoyed solving the overall task	7	**27**
8	The system's instructions were visible long enough for me to read them	-60	**60**
9	The system's instructions sounded robotic	**-20**	20
10	I really wanted to find that trophy	7	**33**
11	The system immediately offered help when I was in trouble	**33**	-60
12	I would recommend this game to a friend	-13	**40**
13	I felt I could trust the system instructions	-13	**40**
14	The system gave me too much information all at once	7	**-60**
15	The system used words and phrases that were easy to understand	**60**	**60**
16	The system's instructions were delivered too early	-53	**-67**
17	The system gave me useful feedback about my progress	**-33**	-47
18	I lost track of time while solving the overall task	**0**	-13
19	I was confused about what to do next	27	**-53**
20	I had no difficulty with identifying the objects the system described for me	47	**80**
21	The system sent instructions too late	40	**-67**
22	The system's instructions were repetitive	**60**	67
23	Overall, the system gave me good directions	27	**60**
	Total winners	8	**16**

successful in Level 3 instructions, since it leads to to ambiguity as described in Section 4.

We conclude that a change in granularity always implies modifying the generator of referring expressions. Additionally, since there are several sources for generating those expressions (e.g. landmarks, relative position with respect to the user, absolute position with respect to the world, temporal relations (e.g. visited/seen before)), it is necessary to have a measure of how good is a referring expression with respect to its *discriminating power* for an certain object or room.

There are many directions for further research. For example, the generation of referring expressions is still an open research problem. Thus, we think that this task could be further developed using a measure (e.g. using Mutual Information or Statistical methods) of the *discriminating power* of a certain combination of attributes according to the object that have to be referred to.

References

1. Benotti L.: Enlightened Update in a Dialogue Game. DECALOG, The 2007 Workshop on the Semantics and Pragmatics of Dialogue. Rovereto, Italy, (2007)
2. Byron, D., Koller, A., Striegnitz,K., Cassell, J., Dale, R., Moore J., Oberlander, J.: Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE), (2009)
3. GIVE Challenge, `http://www.give-challenge.org`
4. Koller, A., Striegnitz,K., Byron, D., Cassell, J., Dale, R., Moore, J., Oberlander, J.: The First Challenge on Generating Instructions in Virtual Environments, (2009)
5. Koller, A., Striegnitz, K., Gargett, A., Byron,D., Cassell, J., Dale, R., Moore,J., Oberlander, J.: Report on the Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2), (2010)
6. Reiter,E., Dale,R.: Building Natural Language Generation Systems. Cambridge University Press (2000)
7. Zue, V., Seneff, S.: Spoken Dialogue Systems. In: Benesty, J., Sondhi, M.M., Huang Y. (eds.) Springer Handbook of Speech Processing, pp. 705–722. Springer Berlin Heidelberg (2008)
8. Zen, H.: Speaker and language adaptive training for HMM-based polyglot speech synthesis. In: Proc. of Interspeech 2010, pp.410–413, Makuhari, Japan, Sept. 2010, (2010)
9. Gales, M., Young, S.: The Application of Hidden Markov Models in Speech Recognition. In: Foundations and Trends in Signal Processing, Vol. 1, No. 3 (2007), pp. 195–304, (2008)