A Natural Language Instructor for pedestrian navigation based in generation by selection

Santiago Avalos
LIIS Group, FaMAF
Universidad Nacional de Córdoba
Córdoba, Argentina
santiagoe.avalos@gmail.com

Luciana Benotti
LIIS Group, FaMAF
Universidad Nacional de Córdoba
Córdoba, Argentina
luciana.benotti@gmail.com

Abstract

In this paper we describe a method for developing a virtual instructor for pedestrian navigation based on real interactions between a human instructor and a human pedestrian. A virtual instructor is an agent capable of fulfilling the role of a human instructor, and its goal is to assist a pedestrian in the accomplishment of different tasks within the context of a real city. The instructor decides what to say using a generation by selection algorithm, based on a corpus of real interactions generated within the world of interest. The instructor is able to react to different requests by the pedestrian. It is also aware of the pedestrian position with a certain degree of uncertainty, and it can use different city landmarks to guide him.

1 Introduction and previous work

Virtual instructors are conversational agents that help a user perform a task. These agents can be useful for many purposes, such as language learning (Nunan, 2004), training in simulated environments (Kim et al., 2009) and entertainment (Dignum, 2012; Jan et al., 2009).

Navigation agents generate verbal route directions for users to go from point A to point B in a given world. The wide variety of techniques to accomplish this task, range from giving complete route directions (all route information in a single instruction), to full interactive dialogue systems which give incremental instructions based on the position of the pedestrian. Although it can recognize pre-established written requests, the instructor presented in this work is not able to interpret utterances from the pedestrian, leaving it unable to generate a full dialogue. The instructor's decisions are based on the pedestrian actual task, his position in the world, and the previous behavior from different human instructors. In order to guide a user while performing a task, an effective instructor must know how to describe what needs to be done in a way that accounts for the nuances of the virtual world and that is enough to engage the trainee or gamer in the activity.

There are two main approaches toward automatically producing instructions. One is the selection approach, in which the task is to pick the appropriate output from a corpus of possible outputs. The other is the composition approach, in which the output is dynamically assembled using some composition procedure, e.g. grammar rules.

The natural language generation algorithm used in this work is a modified version of the generation by selection method described in (Benotti and Denis, 2011).

The advantages of generation by selection are many: it affords the use of complex and human-like sentences, the system is not bound to use written instructions (it may easily use recorded audio clips, for example), and finally, no rule writing by a dialogue expert or manual annotations is needed. The disadvantage of generation by selection is that the resulting dialogue may not be fully coherent (Shawar and Atwell, 2003; Shawar and Atwell, 2005; Gandhe and Traum, 2007).

In previous work, the selection approach to generation has been used in non task-oriented conversational agents such as negotiating agents (Gandhe and Traum, 2007), question answering characters (Leuski et al., 2006) and virtual patients (Kenny et al., 2007). In the work presented in this paper, the conversational agent is task-oriented.

In Section 2 we introduce the framework used in the interaction between the navigation agent and the human pedestrians. We discuss the creation of the human interaction corpus and the method for natural language generation in Section 3; And in Section 4 we explain the evaluation methods and

Proceedings of the of the EACL 2014 Workshop on Dialogue in Motion (DM), pages 33–37,
Gothenburg, Sweden, April 26-30 2014.

the expected results.

2 The GRUVE framework

One of the major problems in developing systems that generate navigation instructions for pedestrians is evaluating them with real users in the real world. This evaluations are expensive, time consuming, and need to be carried out not just at the end of the project but also during the development cycle.

Consequently, there is a need for a common platform to effectively compare the performances of several verbal navigation systems developed by different teams using a variety of techniques.

The GIVE challenge developed a 3D virtual indoor environment for development and evaluation of indoor pedestrian navigation instruction systems (Byron et al., 2007; Koller et al., 2007). In this framework, users walk through a building with rooms and corridors, and interact with the world by pressing buttons. The user is guided by a navigation system that generates route instructions.

The GRUVE framework presented in (Janarthanam et al., 2012) is a web-based environment containing a simulated real world in which users can simulate walking on the streets of real cities whilst interacting with different navigation systems. This system focus on providing a simulated environment where people can look at landmarks and navigate based on spatial and visual instructions provided to them. GRUVE also provides a embedded navigation agent, the Buddy System, which can be used to test the framework. Apart from the virtual environment in which they are based an important difference between GIVE and GRUVE is that, in GRUVE, there is a certain degree of uncertainty about the position of the user.

Figure 1: Snapshot of the GRUVE web-client.

GRUVE presents navigation tasks in a game-world overlaid on top of the simulated real world. The main task consists of a treasure hunting similar to the one presented in GIVE. In our work, we use a modified version of the original framework, in which the main task has been replaced by a set of navigation tasks.

The web-client (see Figure 1) includes an interaction panel that lets the user interact with his navigation system. In addition to user location information, users can also interact with the navigation system using a fixed set or written utterances. The interaction panel provided to the user consists of a GUI panel with buttons and drop-lists which can be used to construct and send requests to the system in form of abstract semantic representations (dialogue actions).

3 The virtual instructor

The virtual instructor is a natural language agent that must help users reach a desired destination within the virtual world. Our method for developing an instructor consists of two phases: an annotation phase and a selection phase. In Section 3.1 we describe the annotation phase. This is performed only once, when the instructor is created, and it consists of automatically generating a corpus formed by associations between each instruction and the reaction to it. In Section 3.2 we describe how the utterance selection is performed every time the virtual instructor generates an instruction.

3.1 Annotation

As described in (Benotti and Denis, 2011), the corpus consists in recorded interactions between two people in two different roles: the Direction Giver (DG), who has knowledge of how to perform the task, and creates the instructions, and the Direction Follower (DF), who travels through the environment following those instructions.

The representation of the virtual world is given by a graph of nodes, each one representing an intersection between two streets in the city. GRUVE provides a planner that can calculate the optimal path from any starting point to a selected destination (this plan consists in the list of nodes the user must travel to reach the desired destination). As the DF user walks through the environment, he cannot change the world that surrounds him. This simplifies the automatic annotation process, and the logged atoms are:

- user position: latitude and longitude, indicating position relative to the world.
- user orientation: angle between 0-360, indicating rotation of the point of view.

In order to define the reaction associated to each utterance, it is enough to consider the position to which the user arrives after an instruction has been given, and before another one is requested. Nine destinations within the city of Edinburgh were selected to be the tasks to complete (the task is to arrive to each destination, from a common starting point, see Figure 2). Each pair of DG and DF had to complete all tasks and record their progress.

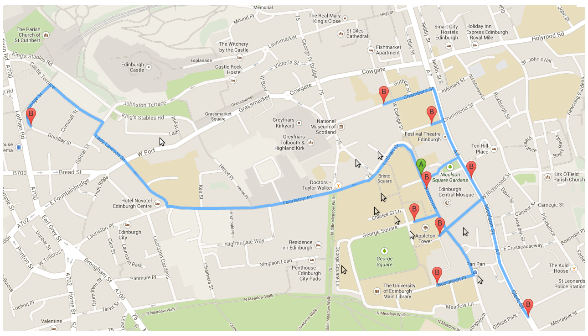

Figure 2: The 9 selected tasks .

For the creation of the corpus, a slightly modified version of the GRUVE wizards-desk was used. This tool is connected to the GRUVE web-client, and allows a human user to act as DF, generating instructions to assist the user in the completion of the task and monitoring his progression. Each instruction generated by a DG was numbered in order, in relation to each task. For example: if the fifth instruction given by the third DG, while performing the second task, was "Go forward and cross the square", then that instruction was numbered as follows:

$$5.3.2 - "Go\ forward\ and\ cross\ the\ square".$$

This notation was included to maintain the generation order between instructions (as the tasks were given in an arbitrary specific order for each DG). With **last-generated**, we refer to the instructions that were generated in the last 3 runs of each DG. This notion is needed to evaluate the effect of the increasing knowledge of the city (this metric is explained in Section 4).

As discussed in (Benotti and Denis, 2011) misinterpreted instructions and corrections result in clearly inappropriate instruction-reaction associations. Since we want to avoid any manual annotation, but we also want to minimize the quantity of errors inside the corpus, we decided to create a first corpus in which the same person portraits the roles of DG and DF. This allows us to eliminate the ambiguity of the instruction interpretation on the DF side, and eliminates correction instructions (instructions that are of no use for guidance, but were made to correct a previous error from the DG, or a wrong action from the DF). Later on, each instruction in this corpus was performed upon the virtual world by various others users, their reactions compared to the original reaction, and scored. For each task, only the instructions whose score exceeded an acceptance threshold remained in the final corpus.

3.2 Instruction selection

The instruction selection algorithm, displayed in Algorithm 1 consists in finding in the corpus the set of candidate utterances C for the current task plan P, which is the sequence of actions that needs to be executed in the current state of the virtual world in order to complete the task. We use the planner included in GRUVE to create P. We define:

$$C = \{U \in Corpus \mid P\ starts\ with\ U.Reaction\}$$

In other words, an utterance U belongs to C if the first action of the current plan P exactly matches the reaction associated to the utterance U. Whenever the plan P changes, as a result of the actions of the DF, we call the selection algorithm in order to regenerate the set of candidate utterances C.

Algorithm 1 Selection Algorithm

$C \leftarrow \emptyset$
$action \leftarrow nextAction(currentObjective)$
for all $Utterance\ U \in Corpus$ **do**
  if $action = U.Reaction$ **then**
    $C \leftarrow C \cup U$
  end if
end for

All the utterances that pass this test are considered paraphrases and hence suitable in the current context. Given a set of candidate paraphrases, one has to consider two cases: the most frequent case when there are several candidates and the possible case when there is no candidate.

- No candidate available: If no instruction is selected because the current plan cannot be matched with any existing reaction, a default, neutral, instruction ”go” is uttered.
- Multiple candidates available: When multiple paraphrases are available, the agent must select one to transmit to the user. In this case, the algorithm selects one from the set of the last-generated instructions for the task (see Section 3.1).

4 Evaluation and expected results

Is this section we present the metrics and evaluation process that will be performed to test the virtual instructor presented in Section 3, which was generated using the dialogue model algorithm introduced in Section 3.2.

4.1 Objective metrics

The objective metrics are summarized below:

- Task success: successful runs.
- Canceled: runs not finished.
- Lost: runs finished but failed.
- Time (sec): average for successful runs.
- Utterances: average per successful run.

With this metrics, we will compare 3 systems: agents A, B and C.

Agent A is the GRUVE buddy system, which is provider by the GRUVE Challenge organizers as a baseline. Agent B consists of our virtual instructor, configured to select a random instruction when presented with multiple candidates (see Section 3.1). Agent C is also our virtual instructor, but when presented with several candidates, C selects a candidate who is also part of the last-generated set. As each task was completed in different order by each DG when the corpus was created, it is expected that in every set of candidates, the most late-generated instructions were created with greater knowledge of the city.

4.2 Subjective metrics

The subjective measures will be obtained from responses to a questionnaire given to each user at the end of the evaluation, based partially on the GIVE-2 Challenge questionnaire (Koller et al., 2010). It ask users to rate different statements about the system using a $0\ to\ 10$ scale.

The questionnaire will include 19 subjective metrics presented below:

Q1: *The system used words and phrases that were easy to understand.*
Q2: *I had to re-read instructions to understand what I needed to do.*
Q3: *The system gave me useful feedback about my progress.*
Q4: *I was confused about what to do next.*
Q5: *I was confused about which direction to go in.*
Q6: *I had no difficulty with identifying the objects the system described for me.*
Q7: *The system gave me a lot of unnecessary Information.*
Q8: *The system gave me too much information all at once.*
Q9: *The system immediately offered help when I was in trouble.*
Q10: *The system sent instructions too late.*
Q11: *The systems instructions were delivered too early.*
Q12: *The systems instructions were clearly worded.*
Q13: *The systems instructions sounded robotic.*
Q14: *The systems instructions were repetitive.*
Q15: *I lost track of time while solving the overall task.*
Q16: *I enjoyed solving the overall task.*
Q17: *Interacting with the system was really annoying.*
Q18: *The system was very friendly.*
Q19: *I felt I could trust the systems instructions.*

Metrics Q1 to Q12 assess the effectiveness and reliability of instructions, while metrics Q13 to Q19 are intended to assess the naturalness of the instructions, as well as the immersion and engagement of the interaction.

4.3 Expected results

Based on the results obtained by (Benotti and Denis, 2011) in the GIVE-2 Challenge, we expect a good rate of successful runs for the agent. Furthermore, the most interesting part of the evaluation resides in the comparison between agents B and C. We expect that the different selection methods of this agents, when presented with multiple instruction candidates, can provide information about the form in which the level of knowledge of the virtual world or environment modifies the capacity of a Direction Giver to create correct, and useful, instructions.

References

Luciana Benotti and Alexandre Denis. 2011. Giving instructions in virtual environments by corpus based selection. In *Proceedings of the SIGDIAL 2011 Conference*, SIGDIAL '11, pages 68–77. Association for Computational Linguistics.

D. Byron, A. Koller, J. Oberlander, L. Stoia, and K. Striegnitz. 2007. Generating instructions in virtual environments (give): A challenge and evaluation testbed for nlg. In *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*.

Frank Dignum. 2012. Agents for games and simulations. *Autonomous Agents and Multi-Agent Systems*, 24(2):217–220, March.

S. Gandhe and D. Traum. 2007. First steps toward dialogue modelling from an un-annotated human-human corpus. In *IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systemss*.

Dusan Jan, Antonio Roque, Anton Leuski, Jacki Morie, and David Traum. 2009. A virtual tour guide for virtual worlds. In *Proceedings of the 9th International Conference on Intelligent Virtual Agents*, IVA '09, pages 372–378, Berlin, Heidelberg. Springer-Verlag.

Srinivasan Janarthanam, Oliver Lemon, and Xingkun Liu. 2012. A web-based evaluation framework for spatial instruction-giving systems. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pages 49–54. Association for Computational Linguistics.

Patrick Kenny, Thomas D. Parsons, Jonathan Gratch, Anton Leuski, and Albert A. Rizzo. 2007. Virtual patients for clinical therapist skills training. In *Proceedings of the 7th International Conference on Intelligent Virtual Agents*, IVA '07, pages 197–210, Berlin, Heidelberg. Springer-Verlag.

Julia M. Kim, Randall W. Hill, Jr., Paula J. Durlach, H. Chad Lane, Eric Forbell, Mark Core, Stacy Marsella, David Pynadath, and John Hart. 2009. Bilat: A game-based environment for practicing negotiation in a cultural context. *Int. J. Artif. Intell. Ed.*, 19(3):289–308, August.

A. Koller, J. Moore, B. Eugenio, J. Lester, L. Stoia, D. Byron, J. Oberlander, and K. Striegnitz. 2007. Shared task proposal: Instruction giving in virtual worlds. In *In Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation.*

Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. Report on the second nlg challenge on generating instructions in virtual environments (give-2). In *Proceedings of the 6th International Natural Language Generation Conference*, INLG '10, pages 243–250. Association for Computational Linguistics.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, SigDIAL '06, pages 18–27. Association for Computational Linguistics.

David Nunan. 2004. *Task-based language teaching*. University Press, Cambridge.

B.A. Shawar and E. Atwell. 2003. Using dialogue corpora to retrain a chatbot system. In *Proceedings of the Corpus Linguistics Conference*, pages 681–690.

B.A. Shawar and E. Atwell. 2005. Using corpora in machine-learning chatbot systems. *International Journal of Corpus Linguistics*, 10:489–516.